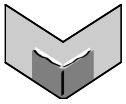
The Journal of Machine Learning Research
Volume 16
Print-Archive Edition

Pages 2611–3909



Microtome Publishing
Brookline, Massachusetts
www.mtome.com

The Journal of Machine Learning Research
Volume 16
Print-Archive Edition

The Journal of Machine Learning Research (JMLR) is an open access journal. All articles published in JMLR are freely available via electronic distribution. This Print-Archive Edition is published annually as a means of archiving the contents of the journal in perpetuity. The contents of this volume are articles published electronically in JMLR in 2015.

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

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

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

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

JMLR Editorial Board

Journal of Machine Learning Research

Volume 16, 2016

Journal of Machine Learning Research 16 (2015) 2611-2616 Submitted 2/14; Revised 12/14; Published 12/15

pyGPs – A Python Library for Gaussian Process Regression and Classification

Marion Neumann M.NEUMANN@WUSTL.EDU
Department of Computer Science and Engineering,
Washington University, St. Louis, MO 63130, United States

Shan Huang SCHAN.HUANG@GMAIL.COM
Fraunhofer IAIS, 53757 Sankt Augustin, Germany

Daniel E. Marthaler DAN.MARTHALER@GMAIL.COM
Sproutling, San Francisco, CA 94111, United States

Kristian Kersting KRISTIAN.KERSTING@CS.TU-DORTMUND.DE
Department of Computer Science, TU Dortmund University
44221 Dortmund, Germany

Editor: Antti Honkela

Abstract

We introduce pyGPs, an object-oriented implementation of Gaussian processes (GPs) for machine learning. The library provides a wide range of functionalities reaching from simple GP specification via mean and covariance and GP inference to more complex implementations of hyperparameter optimization, sparse approximations, and graph based learning. Using Python we focus on usability for both "users" and "researchers". Our main goal is to offer a *user-friendly and flexible* implementation of GPs for machine learning.

Keywords: Gaussian processes, Python, regression and classification

1. Introduction

pyGPs is a Python software project implementing Gaussian processes (GPs) for machine learning (ML). GPs have become a popular model for a wide variety of ML tasks (Rasmussen and Williams, 2006), such as standard regression and classification, as well as active learning (Freytag et al., 2013), graph-based and relational learning (Chu et al., 2006), and Bayesian optimization (Osborne et al., 2009). Besides the recent advances in ML research, GPs get more and more attention for applications in other fields such as animal behaviour research (Mann et al., 2011) or reconfigurable computing (Kurek et al., 2013). Existing procedural GP libraries are GPML (Rasmussen and Nickisch, 2010) and GPstuff (Vanhatalo et al., 2013). However, depending on their design procedural implementations can be hard to extend. Being an established object-oriented programming language Python has great support and is easy to use. There are a few existing Python implementations of GPs. GPs in scikit (Pedregosa et al., 2011) provide only very restricted functionality and they are difficult to extend. pyGP[1] is little developed in terms of documentation and developer interface. GPy (the GPy authors, 2014) was developed in parallel to pyGPs and the library focuses

1. Online at `https://github.com/PMBio/pygp`.

mainly on dimensionality reduction and multi-output learning, whereas our implementation provides extensions for graph-based learning including an implementation of propagation kernels (Neumann et al., 2012), as well as simple routines for multi-class classification, evaluation, and enhanced hyperparameter optimization.

pyGPs is both *user-friendly and flexible*. We explicitly want to bridge the gap between systems designed primarily for "users", who mainly want to apply GPs and need basic ML routines for model training, evaluation, and visualization, and expressive systems for "developers", who focus on extending the core GP functionalities as covariance and likelihood functions, as well as inference techniques. We provide a comprehensive and illustrative documentation including a lot of demos and an overview of functionalities providing an easy start with pyGPs. Further, we believe that utilizing object-oriented programming is the right direction towards our goal of developing user-friendly *and* flexible software.

2. Implementation and Documentation

pyGPs is released under the FreeBSD license and it can be downloaded from `http://mloss.org/software/view/509/` or `https://github.com/marionmari/pyGPs`. pyGPs requires Python 2.6 or 2.7 (`www.python.org`) and the `numpy` (`www.numpy.org`), `scipy` (`www.scipy.org`), and `Matplotlib` (`www.matplotlib.org/`) packages. The provided functionality follows roughly the GPML toolbox introduced in Rasmussen and Nickisch (2010), which is implemented in a procedural way in MATLAB. However, pyGPs has an object-oriented structure and it additionally supports useful routines for the practical use of GPs, such as cross validation functionalities for evaluation as well as basic routines for iterative restarts for GP hyperparameter optimization. The library also supports FITC sparse approximations (Snelson and Ghahramani, 2005), one-vs-one multi-class classification and kernels for graph-based and semi-supervised learning.[2]

pyGPs provides a comprehensive documentation in form of a pdf-manual including an API and an online documentation at `http://www-ai.cs.uni-dortmund.de/weblab/static/api_docs/pyGPs/`. This documentation guides the user through installation, offers a small tutorial on GPs, summarizes the functionalities of the library and walks the user through a lot of demos. There are demo implementations of basic and sparse regression, as well as of basic and sparse binary classification. Further, we show how to do multi-class classification in a one-vs-one fashion, how to perform k-fold cross validation and how to incorporate kernels on graphs and graph kernels. The documentation also gives instructions on how to develop customized kernel, mean, likelihood, or inference functions. pyGPs also includes unit tests and instructions on how to test newly developed functions.

3. Functionalities of pyGPs

Now we exemplify the use of pyGPs for regression and describe its functionalities in detail.

2. We also released the procedural version `pyGP_PR`, which consists of a subset of pyGPs routines and is intended for users familiar with the GPML toolbox. It provides all basic routines needed to follow the examples in Rasmussen and Williams (2006). Online at `https://github.com/marionmari/pyGP_PR`.

3.1 Basic Example

Given the training data (x, y), where $x \in \mathbb{R}^{n \times d}$ and $y \in \mathbb{R}^n$, we get the predictions $f_* = f(z)$ for test inputs $z \in \mathbb{R}^{m \times d}$ by invoking the following four lines:

```
1 model = pyGPs.GPR()      # specify model (GP regression)
2 model.getPosterior(x,y) # get default model (zero mean & rbf kernel)
3 model.optimize(x,y)      # optimize hyperparams (single run minimize)
4 model.predict(z)         # prediction for test cases
```

Besides the predictive mean $\bar{f}_*$ (`model.ym`) of the GP which is commonly used as point estimate for the input targets, the model contains the predictive variance (`model.ys2`) and the means and variances of the latent function (`model.fm` and `model.fs2`).

In the following, we give a more detailed description of the above routine. By specifying the model as GP regression, cf. line 1, we assume a prior GP $f \sim \mathcal{GP}\left(m(x), k(x, x')\right)$, where the default mean function is zero, $m(x) = 0$, and the default covariance is a radial basis function (RBF) kernel, $k(x, x') = \sigma^2 \exp(-\frac{\|x-x'\|^2}{2\ell^2})$, with hyperparameters $\theta = \{\sigma, \ell\}$; both of which have a default value of 1. Further, the default GP regression settings are a Gaussian likelihood function and exact inference. For hyperparameter optimization we use an optimizer introduced in Rasmussen (1996) commonly referred to as *minimize* as the default. We will describe and explain the use of non-default likelihoods, and inference and optimization methods in the next section. Non-default means such as a linear (`mean.Linear`) mean function and covariances such as polynomial (`cov.Poly`) or Matérn (`cov.Matern`) or sums (`+`) and products (`*`) thereof can be set by using `model.setPrior`. A list of implemented means and kernels is provided in Table 1.
The following lines show how to set composite mean and covariance functions:

```
5 m = pyGPs.mean.Linear(D=x.shape[1])+pyGPs.mean.Const()  # sum of means
6 k = pyGPs.cov.RBF() * pyGPs.cov.Linear()           # product of kernels
7 model.setPrior(mean=m, kernel=k)                   # non-default prior
```

After we have specified the GP for regression, we can fit the model to our training data, cf. line 2. Now, we get the current value of the negative log marginal likelihood (`model.nlZ`) and its partial derivatives w.r.t. each hyperparameter (`model.dnlZ`) and the (approximate) posterior (`model.posterior`) represented by $L = \text{cholesky}(K + \sigma_n^2 I)$ (`posterior.L`), $\alpha = L^\top \backslash (L \backslash y)$ (`posterior.alpha`) and σ_n (`posterior.sW`). So far, we performed inference with the default hyperparameters of the specified covariance function. For better results, however, we optimize the hyperparameters, cf. line 3. This means that we minimize the negative log marginal likelihood $-log\ p(y|x, \theta) = -\frac{1}{2}\, y^\top K^{-1} y - \frac{1}{2}\, log|K| - \frac{n}{2}\, log 2\pi$ and fit the model again with the learned hyperparameters. The hyperparameters can be accessed via `model.covfunc.hyp` and the posterior (`model.posterior`) and negative log marginal likelihood (`model.nlZ`) will be updated accordingly. Now, we can get the predictions with the optimal hyperparameters, cf. line 4, where $\bar{f}_*$ is the expected value of $f_*|x, y, z$ (`model.ym`) and $V(f_*)$ is the variance of $f_*|x, y, z$ (`model.ys2`).

3.2 Functionalities

The object-oriented implementation offers one base class `GP` for the general GP model and five base classes for the core GP functionality `Mean`, `Kernel`, `Likelihood`, `Inference`, and `Optimizer`. Tables 1 and 2 show lists of implemented functionalities in pyGPs. Due to

kernels	*kernels for graphs*	*means*	*optimization methods*	*evaluation measures*
CONSTANT	DIFFUSION	CONSTANT	MINIMIZE	ACC
LINEAR (ISO, ARD, ONE)	L+	LINEAR	BFGS	RMSE
RBF (ISO, ISO-UNIT, ARD)	REG LAPLACIAN	ONE	CG	PREC
MATERN (ISO, ARD)	RANDOM WALK	ZERO	SCG	RECALL
RQ (ISO, ARD)	VND			NLPD
PERIODIC	INVERSE COSINE			
POLYNOMIAL	PROPAGATION KERNEL			
PIECWISEPOLY (ISO)				
NOISE				
COMPOSITE: SUM (+), PRODUCT (∗), SCALE (∗)				

Table 1: pyGPs functionality: kernels, means, optimizers, evaluation measures

	likelihood		
inference	GAUSSIAN	LAPLACE	ERROR FUNCTION
EXACT	✓		
LAPLACE	✓		✓
EP	✓	✓	✓
FITC-EXACT	✓		
FITC-LAPLACE	✓		✓
FITC-EP	✓	✓	✓

Table 2: pyGPs functionality: inference methods, likelihoods

the intuitive class hierarchy it is easy to augment the classes by for instance customized covariance functions and likelihoods. This makes pyGPs suitable for researches in ML. Further, we provide functionalities to ease usability of GPs as a machine learning tool as for instance parameter optimization, evaluation, and one-vs-one multi-class classification. They are explained by detailed demos (`demo_GPMC.py`, `demo_Validation.py`) and in the documentation. In the following, we briefly describe the most important aspects of pyGPs.

Sparse Approximations. We support sparse approximations for large scale GPs for regression and classification. We implement the popular "fully independent training conditional" (FITC) approximation (Snelson and Ghahramani, 2005) for exact and approximate inference.

Optimizers. Beside minimize, other optimization methods included in pyGPs are scaled conjugate gradient optimization (SCG) and it is also possible to use built-in optimizers from `scipy` such as conjugate gradient (CG) or the quasi-Newton method BFGS.

Validation. We provide the most common technique for model evaluation, k-fold cross validation (`valid.py`). The implemented evaluation measures are root mean squared error (`RMSE`), accuracy (`ACC`), precision and recall (`Prec`, `Recall`) and the negative log predictive density (`NLPD`) to evaluate the quality of the whole predictive GP model.

GraphExtensions. pyGPs offers the possibility to perform GP inference on networked data. So far, we provide one example graph kernel (propagation kernel (Neumann et al., 2012)), kernels for graph-based and semi-supervised learning, and knn-graph creation.

Currently, we are working on time series modeling and Bayesian optimization with GPs, as well as the incorporation of more state-of-the-art graph kernels for structured data.We also plan to add multi-output GPs, active learning, further application support, and more likelihood and covariance functions in the near future.

Acknowledgments

We would like to thank the following persons for their help in improving this software: Roman Garnett, Maciej Kurek, Hannes Nickisch, Zhao Xu, and Alejandro Molina. This software project is partly supported by the Fraunhofer ATTRACT fellowship STREAM.

References

W. Chu, V. Sindhwani, Z. Ghahramani, and S.S. Keerthi. Relational Learning with Gaussian Processes. In *Advances in Neural Information Processing Systems (NIPS-06)*, pages 289–296. 2006.

A. Freytag, E. Rodner, P. Bodesheim, and J. Denzler. Labeling examples that matter: Relevance-based active learning with gaussian processes. In *Proceedings of the 35th German Conference on Pattern Recognition (GCPR)*, volume 8142 of *Lecture Notes in Computer Science*, pages 282–291. Springer, 2013.

M. Kurek, T. Becker, and W. Luk. Parametric Optimization of Reconfigurable Designs Using Machine Learning. In *Reconfigurable Computing: Architectures, Tools and Applications - 9th International Symposium (ARC-2013)*, pages 134–145, 2013.

R. Mann, R. Freeman, M. A. Osborne, R. Garnett, C. Armstrong, J. Meade, D. Biro, T. Guilford, and S. Roberts. Objectively identifying landmark use and predicting flight trajectories of the homing pigeon using Gaussian processes. *Journal of the Royal Society Interface*, 8(55):210–219, 2011.

M. Neumann, N. Patricia, R. Garnett, and K. Kersting. Efficient Graph Kernels by Randomization. In *Proceedings of the Machine Learning and Knowledge Discovery in Databases - European Conference (ECML/PKDD-12)*, pages 378–393, 2012.

M. A. Osborne, R. Garnett, and S. J. Roberts. Gaussian processes for global optimization. In *Proceedings of the 3rd Learning and Intelligent Optimization Conference (LION-09)*, 2009.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

C. E. Rasmussen. Function minimization using conjugate gradients: Conj, 1996.

C. E. Rasmussen and H. Nickisch. Gaussian Processes for Machine Learning (GPML) Toolbox. *Journal of Machine Learning Research*, 11:3011–3015, 2010.

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

E. Snelson and Z. Ghahramani. Sparse Gaussian Processes using Pseudo-inputs. In *Advances in Neural Information Processing Systems (NIPS-05)*, pages 1257–1264, 2005.

the GPy authors. GPy: A Gaussian process framework in python, 2014. `https://github.com/SheffieldML/GPy`.

J. Vanhatalo, J. Riihimäki, J. Hartikainen, P. Jylänki, V. Tolvanen, and A. Vehtari. Gpstuff: Bayesian modeling with gaussian processes. *Journal of Machine Learning Research*, 14 (1):1175–1179, 2013.

Journal of Machine Learning Research 16 (2015) 2617-2641 Submitted 5/14; Revised 11/14; Published 7/15

Derivative Estimation Based on Difference Sequence via Locally Weighted Least Squares Regression

WenWu Wang WENGEWSH@SINA.COM
Lu Lin LINLU@SDU.EDU.CN
Qilu Securities Institute for Financial Studies & School of Mathematics
Shandong University
Jinan, 250100, China

Editor: Francois Caron

Abstract

A new method is proposed for estimating derivatives of a nonparametric regression function. By applying Taylor expansion technique to a derived symmetric difference sequence, we obtain a sequence of approximate linear regression representation in which the derivative is just the intercept term. Using locally weighted least squares, we estimate the derivative in the linear regression model. The estimator has less bias in both valleys and peaks of the true derivative function. For the special case of a domain with equispaced design points, the asymptotic bias and variance are derived; consistency and asymptotic normality are established. In simulations our estimators have less bias and mean square error than its main competitors, especially second order derivative estimator.

Keywords: nonparametric derivative estimation, locally weighted least squares, bias-correction, symmetric difference sequence, Taylor expansion

1. Introduction

In nonparametric regressions, it is often of interest to estimate mean functions. Many estimation methodologies and relevant theoretical properties have been rigorously investigated, see, for example, Fan and Gijbels (1996), Härdle et al. (2004), and Horowitz (2009). Nonparametric derivative estimation has never attracted much attention as one usually gets the derivative estimates as "by-products" from a local polynomial or spline fit, as Newell and Einbeck (2007) mentioned. However, applications of derivative estimation are important and wide-ranging. For example, in the analysis of human growth data, first and second derivatives of the height as a function of time are important parameters (Müller, 1988; Ramsay and Silverman, 2002): the first derivative has the interpretation of speed and the second derivative acceleration. Another field of application is the change point problems, including exploring the structures of curves (Chaudhuri and Marron, 1999; Gijbels and Goderniaux, 2005), detecting the extremum of derivative (Newell et al., 2005), characterizing submicroscopic nanoparticle (Charnigo et al., 2007) and comparing regression curves (Park and Kang, 2008). Other needs arise in nonparametric regressions themselves, for example, in the construction of confidence intervals (Eubank and Speckman, 1993), in the computation of bias and variance, and in the bandwidth selection (Ruppert et al., 1995).

There are three main approaches of nonparametric derivative estimation in the literature: smoothing spline, local polynomial regression (LPR), and difference-based method. As for smoothing spline, the usual way of estimating derivatives is to take derivatives of spline estimate. Stone (1985) showed that spline derivative estimators achieve the optimal L_2 rate of convergence. Zhou and Wolfe (2000) derived asymptotic bias, variance, and established normality properties. Heckman and Ramsay (2000) considered a penalized version. In the case of LPR, a polynomial obtained by Taylor Theorem is fitted locally by kernel regression. Ruppert and Wand (1994) derived the leading bias and variance terms for general multivariate kernel weights using locally weighted least squares theory. Fan and Gijbels (1996) established its asymptotic properties. Delecroix and Rosa (2007) showed its uniform consistency. In the context of difference-based derivative estimation, Müller et al. (1987) and Härdle (1990) proposed a cross-validation technique to estimate the first derivative by combining difference quotients with kernel smoothing. But the variance of the estimator is proportional to n^2 in the case of equidistant design. Charnigo et al. (2011) employed a variance-reducing linear combination of symmetric quotients called empirical derivative, quantified the asymptotic variance and bias, and proposed a generalized C_p criterion for derivative estimation. De Brabanter et al. (2013) derived L_1 and L_2 rates and established consistency of the empirical derivative.

LPR relies on Taylor expansion—a local approximation, and the main term of Taylor series is the mean rather than the derivatives. The convergence rates of the mean estimation and the derivative estimations are different in LPR. When the mean estimator achieves the optimal rate of convergence, the derivative estimators do not (see Table 3 in Appendix I). Empirical derivative can eliminate the main term of the approximation, but it seems that their asymptotic bias and variance properties have not been well studied. Also large biases may exist in valleys and peaks of the derivative function, and boundary problem caused by estimation variance is still an unsolved problem. Motivated by Tong and Wang (2005) and Lin and Li (2008), we propose a new method to estimate derivatives in the interior. By applying Taylor expansion to a derived symmetric difference sequence, we obtain a sequence of approximate linear regression representation in which the derivative is just the intercept term. Then we estimate the derivative in the linear regression model via locally weighted least squares. The asymptotic bias and variance of the new estimator are derived, consistency and asymptotic normality are established. Theoretical properties and simulation results illustrate that our estimators have less bias, especially higher order derivative estimator. In the theory frame of locally weighted least squares regression, the empirical first derivative is our special case: local constant estimator. In addition, one-side locally weighted least squares regression is proposed to solve the boundary problem of first order derivative estimation.

This paper is organized as follows. Section 2 introduces the motivation and methodology of this paper. Section 3 presents theoretical results of the first order derivative estimator, including the asymptotic bias and variance, consistency and asymptotic normality. Further, we describe the behavior at the boundaries of first order derivative estimation and propose a correction method. Section 4 generalizes the idea to higher order derivative estimation. Simulation studies are given in Section 5, and the paper concludes by some discussions in Section 6. All proofs are given in Appendices A-H, respectively.

2. Motivation and Estimation Methodology for the First Order Derivative

In this section, we first show that where the bias and variance of derivative estimation come from, and then propose a new method for the first order derivative estimation.

2.1 Motivation

Consider the following nonparametric regression model

$$Y_i = m(x_i) + \epsilon_i, \quad 1 \le i \le n, \tag{1}$$

where x_i's are equidistantly designed, that is, $x_i = i/n$, Y_i's are random response variables, $m(\cdot)$ is an unknown smooth mean function, ϵ_i's are independent and identically distributed random errors with $E[\epsilon_i] = 0$ and $Var[\epsilon_i] = \sigma^2$.

If errors ϵ_i's are not present in (1), the model can be expressed as

$$Y_i = m(x_i), \quad 1 \le i \le n. \tag{2}$$

In this case, the observed Y_i's are actually the true values of the mean function at x_i's. Derivative estimation in model (2) can be viewed as a numerical computation problem. Assume that $m(\cdot)$ is three times continuously differentiable on $[0, 1]$. Then Taylor expansions of $m(x_{i\pm j})$ at x_i are given by

$$\begin{aligned} m(x_{i+j}) &= m(x_i) + m^{(1)}(x_i)\frac{j}{n} + \frac{m^{(2)}(x_i)}{2!}\frac{j^2}{n^2} + \frac{m^{(3)}(x_i)}{3!}\frac{j^3}{n^3} + o\left(\frac{j^3}{n^3}\right), \\ m(x_{i-j}) &= m(x_i) - m^{(1)}(x_i)\frac{j}{n} + \frac{m^{(2)}(x_i)}{2!}\frac{j^2}{n^2} - \frac{m^{(3)}(x_i)}{3!}\frac{j^3}{n^3} + o\left(\frac{j^3}{n^3}\right). \end{aligned}$$

In order to eliminate the dominant term $m(x_i)$, we employ a linear combination of $m(x_{i-j})$ and $m(x_{i+j})$ subject to

$$a_{ij} \cdot m(x_{i+j}) + b_{ij} \cdot m(x_{i-j}) = 0 \cdot m(x_i) + 1 \cdot m^{(1)}(x_i) + O\left(\frac{j}{n}\right).$$

It is equivalent to solving the equations

$$\begin{cases} a_{ij} + b_{ij} = 0, \\ (a_{ij} - b_{ij})\dfrac{j}{n} = 1, \end{cases}$$

whose solution is

$$\begin{cases} a_{ij} = \dfrac{n}{2j}, \\ b_{ij} = -\dfrac{n}{2j}. \end{cases}$$

So we obtain

$$m^{(1)}(x_i) = \frac{m(x_{i+j}) - m(x_{i-j})}{2j/n} - \frac{m^{(3)}(x_i)}{6}\frac{j^2}{n^2} + o\left(\frac{j^2}{n^2}\right). \tag{3}$$

As j increases , the bias will also increase. To minimize the bias, set $j = 1$. Then the first order derivative $m^{(1)}(x_i)$ is estimated by

$$\hat{m}^{(1)}(x_i) = \frac{m(x_{i+1}) - m(x_{i-1})}{2/n}.$$

Here the estimation bias is only the remainder term in Taylor expansion.

We now consider the true regression model (1). Symmetric (about i) difference quotients (Charnigo et al., 2011; De Brabanter et al., 2013) are defined as

$$Y_{ij}^{(1)} = \frac{Y_{i+j} - Y_{i-j}}{x_{i+j} - x_{i-j}}, \quad 1 \le j \le k, \tag{4}$$

where k is a positive integer. Under model (1), we can decompose $Y_{ij}^{(1)}$ into two parts as

$$Y_{ij}^{(1)} = \frac{m(x_{i+j}) - m(x_{i-j})}{2j/n} + \frac{\epsilon_{i+j} - \epsilon_{i-j}}{2j/n}, \quad 1 \le j \le k. \tag{5}$$

On the right hand side of (5), the first term includes the bias, and the second term contains the information of the variance.

From (3) and (5), we have

$$Y_{ij}^{(1)} = m^{(1)}(x_i) + \frac{m^{(3)}(x_i)}{6}\frac{j^2}{n^2} + o\left(\frac{j^2}{n^2}\right) + \frac{\epsilon_{i+j} - \epsilon_{i-j}}{2j/n}. \tag{6}$$

Taking expectation on (6), we have

$$\begin{aligned} E[Y_{ij}^{(1)}] &= m^{(1)}(x_i) + \frac{m^{(3)}(x_i)}{6}\frac{j^2}{n^2} + o\left(\frac{j^2}{n^2}\right) \\ &\doteq m^{(1)}(x_i) + \frac{m^{(3)}(x_i)}{6}\frac{j^2}{n^2}. \end{aligned}$$

For any fixed $k = o(n)$,

$$E[Y_{ij}^{(1)}] \doteq m^{(1)}(x_i) + \frac{m^{(3)}(x_i)}{6} d_j, \quad 1 \le j \le k, \tag{7}$$

where $d_j = \frac{j^2}{n^2}$. We treat (7) as a linear regression with d_j and $Y_{ij}^{(1)}$ as the independent variable and dependent variable respectively, and then estimate $m^{(1)}(x_i)$ as the intercept using the locally weighted least squares regression.

2.2 Estimation Methodology

For a fixed x_i, express equation (6) in the following form:

$$Y_{ij}^{(1)} = \beta_{i0} + \beta_{i1} d_{1j} + \delta_{ij}, \quad 1 \le j \le k,$$

where $\beta_{i0} = m^{(1)}(x_i)$, $\beta_{i1} = \frac{m^{(3)}(x_i)}{6}$, $d_{1j} = \frac{j^2}{n^2}$, and $\delta_{ij} = o\left(\frac{j^2}{n^2}\right) + \frac{\epsilon_{i+j} - \epsilon_{i-j}}{2j/n}$ are independent across j. The above expression takes a regression form, in which the independent variable is d_{1j} and the dependent variable $Y_{ij}^{(1)}$, and the error term satisfies

$$E[\delta_{ij}] = o\left(\frac{j^2}{n^2}\right) \doteq 0, \quad Var[\delta_{ij}] = \frac{n^2\sigma^2}{2j^2}.$$

To reduce the variance and combine the information for all j, we use the locally weighted least squares regression (LWLSR) to estimate coefficients as

$$\begin{aligned}\hat{\beta}_i &= \arg\min_{\beta_{i0},\beta_{i1}} \sum_{j=1}^{k} (Y_{ij}^{(1)} - \beta_{i0} - \beta_{i1} d_{1j})^2 w_{ij} \\ &= (D^\top W D)^{-1} D^\top W Y_i^{(1)},\end{aligned}$$

where $w_{ij} = \frac{\sigma^2/2}{Var[\delta_{ij}]} = \frac{j^2}{n^2}$, $\hat{\beta}_i = (\hat{\beta}_{i0}, \hat{\beta}_{i1})^\top$, superscript $\top$ denotes the transpose of a matrix,

$$D = \begin{pmatrix} 1 & 1^2/n^2 \\ 1 & 2^2/n^2 \\ \vdots & \vdots \\ 1 & k^2/n^2 \end{pmatrix}, Y_i^{(1)} = \begin{pmatrix} Y_{i1}^{(1)} \\ Y_{i2}^{(1)} \\ \vdots \\ Y_{ik}^{(1)} \end{pmatrix}, W = \begin{pmatrix} 1^2/n^2 & 0 & \cdots & 0 \\ 0 & 2^2/n^2 & \cdots & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & 0 & \cdots & k^2/n^2 \end{pmatrix}.$$

Therefore, the estimator is obtained as

$$\hat{m}^{(1)}(x_i) = \hat{\beta}_{i0} = e_1^\top \hat{\beta}_i, \tag{8}$$

where $e_1 = (1, 0)^\top$.

3. Properties of the First Order Derivative Estimation

In this section, we study asymptotic properties of our first order derivative estimator (8) in interior points, and reveal that empirical first derivative is our special case: local constant estimator. For boundary points, we propose one-side LWLSR to reduce estimation variance.

3.1 Asymptotic Results

The following theorems provide asymptotic results on bias and variance, and establish pointwise consistency and asymptotic normality of the first order derivative estimators.

Theorem 1 (**Uniform Asymptotic Variance**) *Assume that the nonparametric model* (1) *holds with equidistant design and the unknown smooth function* $m(\cdot)$ *is three times continuously differentiable on* $[0, 1]$. *Furthermore, assume that the third order derivative* $m^{(3)}(\cdot)$ *is finite on* $[0, 1]$. *Then the variance of the first order derivative estimator in* (8) *is*

$$Var[\hat{m}^{(1)}(x_i)] = \frac{75\sigma^2}{8}\frac{n^2}{k^3} + o\left(\frac{n^2}{k^3}\right)$$

uniformly for $k + 1 \leq i \leq n - k$.

Theorem 1 shows that the variance of the derivative estimator is constant as x changes, while the following theorem shows that the bias changes with x.

Theorem 2 (**Pointwise Asymptotic Bias**) *Assume that the nonparametric model* (1) *holds with equidistant design and the unknown smooth function $m(\cdot)$ is five times continuously differentiable on $[0,1]$. Furthermore, assume that the fifth order derivative $m^{(5)}(\cdot)$ is finite on $[0,1]$. Then the bias of the first order derivative estimator in* (8) *is*

$$Bias[\hat{m}^{(1)}(x_i)] = -\frac{m^{(5)}(x_i)}{504}\frac{k^4}{n^4} + o\left(\frac{k^4}{n^4}\right)$$

for $k+1 \leq i \leq n-k$.

Using Theorems 1 and 2, we have that if $nk^{-3/2} \to 0$ and $n^{-1}k \to 0$, then our estimator has the consistency property

$$\hat{m}^{(1)}(x_i) \xrightarrow{P} m^{(1)}(x_i).$$

Furthermore, we establish asymptotic normality in the following theorem.

Theorem 3 (**Asymptotic Normality**) *Under the assumptions of Theorem 2, if $k \to \infty$ as $n \to \infty$ such that $nk^{-3/2} \to 0$ and $n^{-1}k \to 0$, then*

$$\frac{k^{3/2}}{n}\left(\hat{m}^{(1)}(x_i) - m^{(1)}(x_i) + \frac{m^{(5)}(x_i)}{504}\frac{k^4}{n^4}\right) \xrightarrow{d} N\left(0, \frac{75\sigma^2}{8}\right)$$

for $k+1 \leq i \leq n-k$. Further, if $k \to \infty$ as $n \to \infty$ such that $nk^{-3/2} \to 0$ and $n^{-1}k^{11/10} \to 0$, then

$$\frac{k^{3/2}}{n}\left(\hat{m}^{(1)}(x_i) - m^{(1)}(x_i)\right) \xrightarrow{d} N\left(0, \frac{75\sigma^2}{8}\right)$$

for $k+1 \leq i \leq n-k$.

Theorem 3 shows that with suitable choice of k our first order derivative estimator is asymptotically normally distributed, even asymptotically unbiased. Using the asymptotic normality property, we can construct confidence intervals and confidence bands. From the above theorems, the following corollary follows naturally.

Corollary 4 *Under the assumptions of Theorem 2, the optimal choice of k that minimizes the asymptotic mean square error of the first order derivative estimator in* (8) *is*

$$k_{opt} \doteq 3.48\left(\frac{\sigma^2}{(m^{(5)}(x_i))^2}\right)^{1/11} n^{10/11}.$$

With the optimal choice of k, the asymptotic mean square error of the first order derivative estimator in (8) *can be expressed as*

$$AMSE[\hat{m}^{(1)}(x_i)] \doteq 0.31\left(\sigma^{16}(m^{(5)}(x_i))^6\right)^{1/11} n^{-8/11}.$$

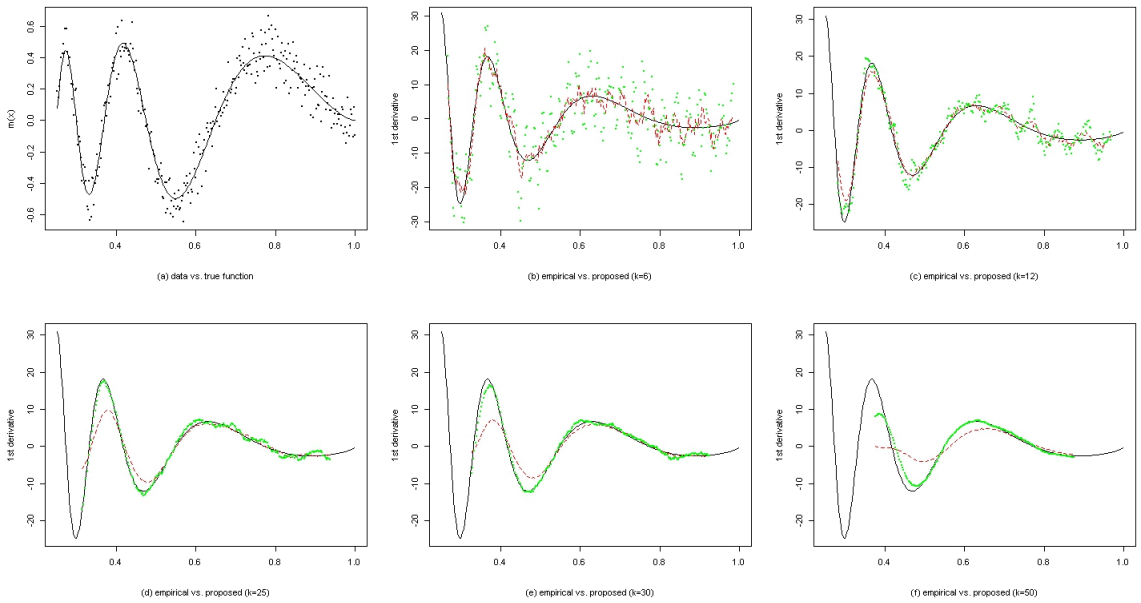


Figure 1: (a) Simulated data set of size 300 from model (1) with equidistant $x_i \in [0.25, 1]$, $m(x) = \sqrt{x(1-x)}\sin((2.1\pi)/(x+0.05))$, $\epsilon_i \overset{iid}{\sim} N(0, 0.1^2)$, and the true mean function (bold line). (b)-(f) The proposed first order derivative estimators (green dots) and the empirical first derivatives (red dashed lines) for $k \in \{6, 12, 25, 30, 50\}$. As a reference, the true first order derivative is also plotted (bold line).

Now we briefly examine the finite sample behavior of our estimator and compare it with the empirical first derivative given by Charnigo et al. (2011) and De Brabanter et al. (2013). Their estimator has the following form:

$$Y_i^{[1]} = \sum_{j=1}^{k_1} w_{ij} Y_{ij}^{(1)}, \quad k_1 + 1 \le i \le n - k_1, \tag{9}$$

where k_1 is a positive integer, $w_{ij} = \frac{j^2/n^2}{\sum_{j=1}^{k_1} j^2/n^2}$, and $Y_{ij}^{(1)}$ is defined in (4).

Figure 1 displays our proposed first order derivative estimators (8) and empirical first derivatives (9) with $k_1 = k \in \{6, 12, 25, 30, 50\}$, for a data set of size 300 generated from model (1) with $x_i \in [0.25, 1]$, $\epsilon_i \overset{iid}{\sim} N(0, 0.1^2)$, and $m(x) = \sqrt{x(1-x)}\sin((2.1\pi)/(x+0.05))$. This $m(x)$ is borrowed from De Brabanter et al. (2013). When k is small (see Figure 1 (b) and (c)), the proposed estimators are noise corrupted versions of the true first order derivatives, while the performance of the empirical derivatives is better except that there are large biases near local peaks and valleys of the true derivative function. As k becomes bigger (see Figure 1 (d)- (f)), our estimators have much less biases than empirical derivative estimators near local peaks and valleys of the true derivative. The balance between the

estimation bias and variance is clear even visually. Furthermore, if we combine the left part of Figure 1 (d), the middle part of (e) and the right part of (f), more accurate derivative estimators are obtained.

Actually, empirical first derivative and our estimator have a close relationship. Express equation (6) in simple linear regression form

$$Y_{ij}^{(1)} = \beta_{i0} + \eta_{ij}, \quad 1 \leq j \leq k,$$

where $\beta_{i0} = m^{(1)}(x_i)$, $\eta_{ij} = O\left((\frac{j}{n})^2\right) + \frac{\epsilon_{i+j} - \epsilon_{i-j}}{2j/n}$ with

$$E[\eta_{ij}] \doteq 0, \quad Var[\eta_{ij}] = \frac{n^2\sigma^2}{2j^2}.$$

This is called the local constant-truncated estimator. By the LWLSR, we get

$$\hat{m}^{(1)}(x_i) = Y_i^{[1]},$$

which is exactly the empirical first derivative. On three times continuous differentiability, we have the following bias and variance

$$Bias[Y_i^{[1]}] = \frac{m^{(3)}(x_i)}{10}\frac{k^2}{n^2}, \quad Var[Y_i^{[1]}] = \frac{3\sigma^2}{2}\frac{n^2}{k^3}.$$

For empirical first derivative and our estimator, symmetric difference sequence eliminates the even-order terms in Taylor expansion of mean function. This is an important advantage, i.e., if the mean function is two times continuously differentiable, then the second-order term is eliminated so that the bias is smaller than the second-order term. In De Brabanter et al. (2013), the bias is $O(k/n)$ which is obtained via a inequality (See Appendix B, De Brabanter et al. 2013). In fact, the bias should be

$$Bias[Y_i^{[1]}] < O(k/n) \quad or \quad Bias[Y_i^{[1]}] = o(k/n),$$

which does not have exact and explicit expression on two times continuous differentiability. In order to obtain explicit expression, we make the stronger smoothing condition—three times continuous differentiability.

In addition, the smoothing assumptions on bias and variance are different in Theorem 1 and Theorem 2. For empirical first derivative and ours, the variance term only needs one time continuous differentiability; whereas the bias term needs three times and five times respectively. From the viewpoint of Taylor expansion, it seems we pay a serious price. However, in practical applications bias-correction is needed especially in the cases of immense oscillation of mean function. From the viewpoint of *Weierstrass approximation theorem,* even if a continuous function is nondifferentiable we still can correct the bias.

3.2 Behavior at the Boundaries

Recall that for the boundary region ($2 \leq i \leq k$ and $n - k + 1 \leq i \leq n - 1$) the weights in empirical first derivative (9) are slightly modified by normalizing the weight sum. Whereas

our estimator can be obtained directly from the LWLSR without any modification, the only difference is that the smoothing parameter is $i-1$ instead of k.

For the boundary ($3 \le i \le k$), the bias and variance for our estimator are

$$Bias[\hat{m}^{(1)}(x_i)] = -\frac{m^{(5)}(x_i)}{504}\frac{(i-1)^4}{n^4}, \quad Var[\hat{m}^{(1)}(x_i)] = \frac{75\sigma^2}{8}\frac{n^2}{(i-1)^3}. \tag{10}$$

Hence, the variance will be the largest for $i = 3$ and decrease for growing i till $i = k$, whereas the bias will be smallest for $i = 3$ and increase for growing i till $i = k$. A similar analysis for $n-k+1 \le i \le n-2$ shows the same results.

For the modified estimator (De Brabanter et al., 2013), the bias and variance in the theory frame of the LWLSR are

$$Bias[\hat{m}^{(1)}(x_i)] = \frac{m^{(3)}(x_i)}{10}\frac{(i-1)^2}{n^2}, \quad Var[\hat{m}^{(1)}(x_i)] = \frac{3\sigma^2}{2}\frac{n^2}{(i-1)^3},$$

Which have the analogue change trend like (10) above. Although our estimator has less bias ($O(1/n^4)$) than empirical first derivative ($O(1/n^2)$), the variances both are big enough ($O(n^2)$). So the two estimators are inaccurate and the boundary problem still exists.

In order to reduce the variance, we propose the one-side locally weighted least squares regression method which consists of two cases: left-side locally weighted least squares regression (LSLWLSR) and right-side locally weighted least squares regression (RSLWLSR). These estimation methods can be used for the boundary: LSLWLSR is for $n-k+1 \le i \le n$ and RSLWLSR is for $1 \le i \le k$. On two times continuous differentiability, the estimation bias is $O(k/n)$ and variance is $O(n^2/k^3)$.

Assume that $m(\cdot)$ is two times continuously differentiable on $[0,1]$. For $1 \le i \le n-k$, define right-side lag-j first-order difference sequence

$$Y_{ij}^{<1>} = \frac{Y_{i+j} - Y_i}{x_{i+j} - x_i}, \quad 1 \le j \le k. \tag{11}$$

Decompose $Y_{ij}^{<1>}$ into two parts and simplify from (11) such as

$$\begin{aligned} Y_{ij}^{<1>} &= \frac{m(x_{i+j}) - m(x_i)}{j/n} + \frac{\epsilon_{i+j} - \epsilon_i}{j/n} \\ &= m^{(1)}(x_i) + \frac{m^{(2)}(x_i)}{2!}\frac{j^1}{n^1} + o\left(\frac{j^1}{n^1}\right) + \frac{\epsilon_{i+j} - \epsilon_i}{j/n}. \end{aligned} \tag{12}$$

For some fixed i, ϵ_i is constant as j increases. Thus we express equation (12) in the following form:

$$Y_{ij}^{<1>} = \beta_{i0} + \beta_{i1} d_{1j} + \delta_{ij}, \quad 1 \le j \le k,$$

where $\beta_{i0} = m^{(1)}(x_i)$, $\beta_{i1} = -\epsilon_i$, $d_{1j} = \frac{n}{j}$, and $\delta_{ij} = \frac{m^{(2)}(x_i)}{2!}\frac{j^1}{n^1} + o\left(\frac{j^1}{n^1}\right) + \frac{\epsilon_{i+j}}{j/n}$ are independent across j with

$$E[\delta_{ij}|\epsilon_i] = \frac{m^{(2)}(x_i)}{2!}\frac{j^1}{n^1} + o\left(\frac{j^1}{n^1}\right) \doteq 0, \quad Var[\delta_{ij}|\epsilon_i] = \frac{n^2\sigma^2}{j^2}.$$

So we use the LWLSR to estimate regression coefficients as

$$\begin{aligned}\hat{\beta}_i &= \arg\min_{\beta_{i0},\beta_{i1}} \sum_{j=1}^{k} (Y_{ij}^{(1)} - \beta_{i0} - \beta_{i1} d_{1j})^2 w_{ij} \\ &= (D^\top W D)^{-1} D^\top W Y_i^{<1>},\end{aligned}$$

where $w_{ij} = \frac{\sigma^2}{Var[\delta_{ij}]} = \frac{j^2}{n^2}$, $\hat{\beta}_i = (\hat{\beta}_{i0}, \hat{\beta}_{i1})^\top$,

$$D = \begin{pmatrix} 1 & n^1/1^1 \\ 1 & n^1/2^1 \\ \vdots & \vdots \\ 1 & n^1/k^1 \end{pmatrix}, Y_i^{<1>} = \begin{pmatrix} Y_{i1}^{<1>} \\ Y_{i2}^{<1>} \\ \vdots \\ Y_{ik}^{<1>} \end{pmatrix}, W = \begin{pmatrix} 1^2/n^2 & 0 & \cdots & 0 \\ 0 & 2^2/n^2 & \cdots & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & 0 & \cdots & k^2/n^2 \end{pmatrix}.$$

Therefore, the estimator is obtained as

$$\hat{m}^{(1)}(x_i) = \hat{\beta}_{i0} = e_1^\top \hat{\beta}_i, \tag{13}$$

where $e_1 = (1, 0)^\top$.

Following Theorem 1-4 above, we have the similar theorems for the right-side first-order derivative estimator in (13). Here we only give the asymptotic bias and variance as follows.

Theorem 5 *Assume that the nonparametric model (1) holds with equidistant design and the unknown smooth function $m(\cdot)$ is two times continuously differentiable on $[0, 1]$. Furthermore, assume that the second order derivative $m^{(2)}(\cdot)$ is finite on $[0, 1]$. Then the bias and variance of the right-side first-order derivative estimator in* (13) *are*

$$\begin{aligned} Bias[\hat{m}^{(1)}(x_i)|\epsilon_i] &= \frac{m^{(2)}(x_i)}{2} \frac{k^1}{n^1} + o\left(\frac{k^1}{n^1}\right) \\ Var[\hat{m}^{(1)}(x_i)|\epsilon_i] &= 12\sigma^2 \frac{n^2}{k^3} + o\left(\frac{n^2}{k^3}\right) \end{aligned}$$

correspondingly for $1 \le i \le n - k$.

From Theorem 5 above, we can see that the variance and bias for the right-side first-order derivative estimator in (13) is $O(n^2/k^3)$ and $O(k/n)$, which is the same rate as De Brabanter et al. (2013) deduced on two times continuous differentiability. For further bias-correction, high-order Taylor expansion may be needed. A similar analysis for left-side lag-j first-order difference sequence obtains the same results.

3.3 The Choice of k

From the tradeoff between bias and variance, we have two methods for the choice of k: adaptive method and uniform method. The adaptive k based on asymptotic mean square error is

$$k_a = 3.48 \left(\frac{\sigma^2}{(m^{(5)}(x_i))^2}\right)^{1/11} n^{10/11}.$$

To choose k globally, we consider the mean averaged square error (MASE) criterion

$$\begin{aligned} MASE &= \frac{1}{n-2k}\sum_{i=k+1}^{n-k} MSE(\hat{m}^{(1)}(x_i)) \\ &= \frac{1}{n-2k}\sum_{i=k+1}^{n-k}(\frac{75\sigma^2}{8}\frac{n^2}{k^3}+\frac{(m^{(5)}(x_i))^2}{504^2}\frac{k^8}{n^8}) \\ &= \frac{75\sigma^2}{8}\frac{n^2}{k^3}+\frac{1}{n-2k}\sum_{i=k+1}^{n-k}\frac{(m^{(5)}(x_i))^2}{504^2}\frac{k^8}{n^8} \\ &\to \frac{75\sigma^2}{8}\frac{n^2}{k^3}+\frac{L_5}{504^2}\frac{k^8}{n^8}, \end{aligned}$$

where $L_5 = \int_0^1 (m^{(5)}(x))^2 \mathrm{d}x$. Minimizing the $MASE$ with respect to k, the uniform k is

$$k_u = 3.48(\frac{\sigma^2}{L_5})^{1/11} n^{10/11}.$$

Since the k_a and k_u are unknown in practice, a rule of thumb estimator may be preferable. The error variance σ^2 can be estimated by Hall et al. (1990), the fifth order derivative $m^{(5)}(x_i)$ can be estimated by local polynomial regression (R-package: locpol), and L_5 is estimated by Seifert et al. (1993).

However, questions still remain. First, $k = O(n^{10/11})$, which requires n to be large enough to ensure $k < n$; Second, 'the higher the derivative, the wilder the behavior' (Ramsay, 1998), thus the estimations of $m^{(5)}(x_i)$ and L_5 are inaccurate. The most important is that when the bias is very small or large we can't balance the bias and variance via only increasing or decreasing the value of k. From the expression of adaptive k, uniform k and simulations, we put forward the following considerations.

- On the whole, k should satisfy $k < n/2$ or else the needed data size $2k$ goes over the total size n so that we can't estimate any derivative. In addition, we can't leave more boundary points than interior points, so k needs to satisfy the condition $k < n/4$.

- The choice of k relies on Taylor expansion which is a local concept. There exists some maximum value of k suitable for a fixed mean function, denoted by k_{max}. However, adaptive and uniform k is determined by many factor: variance, sample size, frequency and amplitude of mean function. Thus it is possible to obtain too big k in the cases of large variance, and now cross validation could be an alternative. As frequency and amplitude increase, the uniform and adaptive k decrease. *This is the reason why our estimator adopting different k for different oscillation has better performance in Figure 1.* In addition, as the order of Taylor expansion increases, the k_{max} becomes large. So our estimator needs a larger k than empirical derivative.

- When the third-order and fifth-order derivatives are close to zero, the values of k_a and k_u are too big even $k > n/2$. Thus we can't balance bias and variance via increasing the value of k when bias is very small. Meanwhile we can't balance bias and variance via decreasing the value of k when bias is too big. It is better to correct bias by higher-order Taylor expansion.

4. Higher Order Derivative Estimations

In this section, we generalize the idea of the first order derivative estimation to higher order. Different difference sequences are adopted for first and second order derivative estimation.

4.1 Second Order Derivative Estimation

As for the second order derivative estimation, we can show by a similar technique as in (3) that

$$m^{(2)}(x_i) = \frac{m(x_{i-j}) - 2m(x_i) + m(x_{i+j})}{j^2/n^2} - \frac{m^{(4)}(x_i)}{12}\frac{j^2}{n^2} + o\left(\frac{j^2}{n^2}\right).$$

Define

$$Y_{ij}^{(2)} = \frac{Y_{i-j} - 2Y_i + Y_{i+j}}{j^2/n^2}. \tag{14}$$

Just as in equation (5), decompose (14) into two parts as

$$Y_{ij}^{(2)} = \frac{m(x_{i-j}) - 2m(x_i) + m(x_{i+j})}{j^2/n^2} + \frac{\epsilon_{i-j} - 2\epsilon_i + \epsilon_{i+j}}{j^2/n^2}, \quad 1 \le j \le k.$$

Note that i is fixed as j changes. Thus the conditional expectation of $Y_{ij}^{(2)}$ given ϵ_i is

$$\begin{aligned} E[Y_{ij}^{(2)}|\epsilon_i] &= \frac{m(x_{i-j}) - 2m(x_i) + m(x_{i+j})}{j^2/n^2} + (-2\epsilon_i)\frac{n^2}{j^2} \\ &\doteq m^{(2)}(x_i) + \frac{m^{(4)}(x_i)}{12}\frac{j^2}{n^2} + (-2\epsilon_i)\frac{n^2}{j^2}. \end{aligned}$$

Therefore, the new regression model is given by

$$Y_{ij}^{(2)} = \beta_{i0} + \beta_{i1}d_{1j} + \beta_{i2}d_{2j} + \delta_{ij}, \quad 1 \le j \le k,$$

where the regression coefficient vector $\beta_i = (\beta_{i0}, \beta_{i1}, \beta_{i2})^\top = (m^{(2)}(x_i), \frac{m^{(4)}(x_i)}{12}, -2\epsilon_i)^\top$, covariates $d_{1j} = \frac{j^2}{n^2}$ and $d_{2j} = \frac{n^2}{j^2}$, and the error term $\delta_{ij} = \frac{\epsilon_{i+j}+\epsilon_{i-j}}{j^2/n^2} + o\left(\frac{j^2}{n^2}\right)$, with

$$E[\delta_{ij}|\epsilon_i] \doteq 0, \quad Var[\delta_{ij}|\epsilon_i] = \frac{2\sigma^2 n^4}{j^4}.$$

Now the locally weighted least squares estimator of β_i can be expressed as

$$\hat{\beta}_i = (D^\top W D)^{-1} D^\top W Y_i^{(2)},$$

where

$$D = \begin{pmatrix} 1 & 1^2/n^2 & n^2/1^2 \\ 1 & 2^2/n^2 & n^2/2^2 \\ \vdots & \vdots & \vdots \\ 1 & k^2/n^2 & n^2/k^2 \end{pmatrix}, Y_i^{(2)} = \begin{pmatrix} Y_{i1}^{(2)} \\ Y_{i2}^{(2)} \\ \vdots \\ Y_{ik}^{(2)} \end{pmatrix}, W = \begin{pmatrix} 1^4/n^4 & 0 & \cdots & 0 \\ 0 & 2^4/n^4 & \cdots & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & 0 & \cdots & k^4/n^4 \end{pmatrix}.$$

Therefore,

$$\hat{m}^{(2)}(x_i) = \hat{\beta}_{i0} = e_1^\top \hat{\beta}_i, \tag{15}$$

where $e_1 = (1, 0, 0)^\top$.

The following three theorems provide asymptotic results on bias, variance and mean square error, and establish pointwise consistency and asymptotic normality of the second order derivative estimator.

Theorem 6 *Assume that the nonparametric model* (1) *holds with equidistant design and the unknown smooth function $m(\cdot)$ is six times continuously differentiable on $[0, 1]$. Furthermore, assume that the sixth order derivative $m^{(6)}(\cdot)$ is finite on $[0, 1]$. Then the variance of the second order derivative estimator in* (15) *is*

$$Var[\hat{m}^{(2)}(x_i)|\epsilon_i] = \frac{2205\sigma^2}{8}\frac{n^4}{k^5} + o(\frac{n^4}{k^5})$$

uniformly for $k + 1 \leq i \leq n - k$, and the bias is

$$Bias[\hat{m}^{(2)}(x_i)|\epsilon_i] = -\frac{m^{(6)}(x_i)}{792}\frac{k^4}{n^4} + o(\frac{k^4}{n^4})$$

for $k + 1 \leq i \leq n - k$.

From Theorem 6, we can see that if $nk^{-5/4} \to 0$ and $n^{-1}k \to 0$, then our estimator is consistent

$$\hat{m}^{(2)}(x_i) \xrightarrow{P} m^{(2)}(x_i).$$

Moreover, we establish asymptotic normality, derive the asymptotic mean square error and the optimal k value.

Corollary 7 *Under the assumptions of Theorem 6, if $k \to \infty$ as $n \to \infty$ such that $nk^{-5/4} \to 0$ and $n^{-1}k \to 0$, then*

$$\frac{k^{5/2}}{n^2}\left(\hat{m}^{(2)}(x_i) - m^{(2)}(x_i) + \frac{m^{(6)}(x_i)}{792}\frac{k^4}{n^4}\right) \xrightarrow{d} N\left(0, \frac{2205\sigma^2}{8}\right).$$

Moreover, if $nk^{-5/4} \to 0$ and $n^{-1}k^{13/12} \to 0$, then

$$\frac{k^{5/2}}{n^2}\left(\hat{m}^{(2)}(x_i) - m^{(2)}(x_i)\right) \xrightarrow{d} N\left(0, \frac{2205\sigma^2}{8}\right).$$

Corollary 8 *Under the assumptions of Theorem 6, the optimal k value that minimizes the asymptotic mean square error of the second order derivative estimator in* (15) *is*

$$k_{opt} \doteq 4.15\left(\frac{\sigma^2}{(m^{(6)}(x_i))^2}\right)^{1/13} n^{12/13}.$$

With the optimal choice of k, the asymptotic mean square error of the second order derivative estimator in (15) *can be expressed as*

$$AMSE[\hat{m}^{(1)}(x_i)] \doteq 0.36\left(\sigma^{16}(m^{(6)}(x_i))^{10}\right)^{1/13} n^{-8/13}.$$

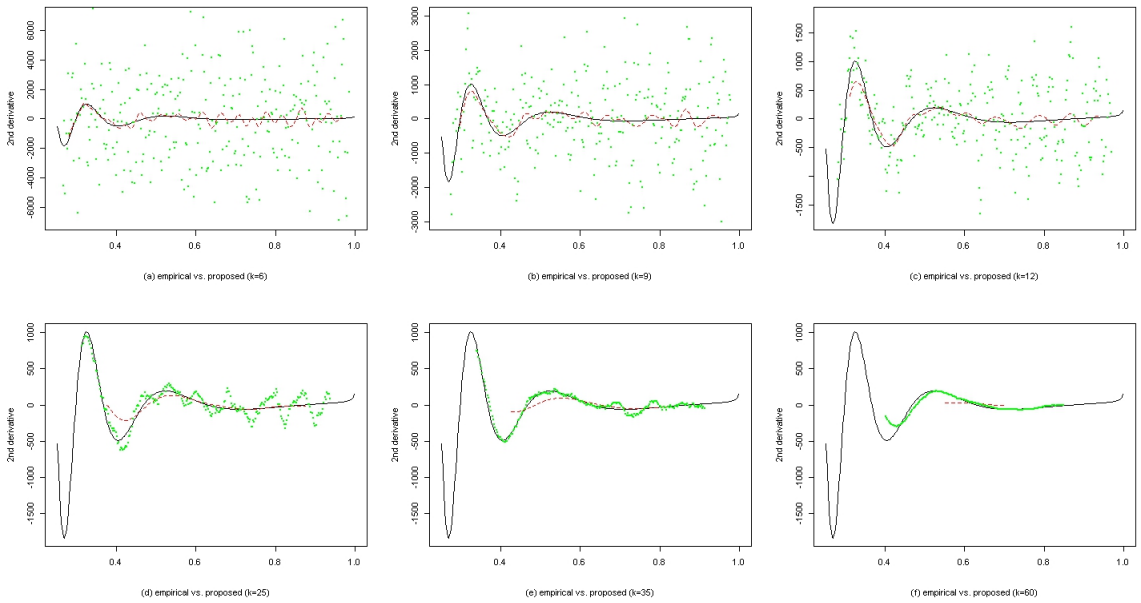


Figure 2: (a)-(f) The proposed second order derivative estimators (green points) and the empirical second derivatives (red dashed line) for $k \in \{6, 9, 12, 25, 35, 60\}$ based on the simulated data set from Figure 1. As a reference, the true second order derivative curve is also plotted (bold line).

Here we also use a simple simulation to examine the finite sample behavior of the new estimator and compare it with the empirical second derivative given by Charnigo et al. (2011) and De Brabanter et al. (2013). Their estimator has the following form:

$$Y_i^{[2]} = \sum_{j=1}^{k_2} w_{ij} Y_{ij}^{(2)}, \quad k_1 + k_2 + 1 \leq i \leq n - k_1 - k_2, \tag{16}$$

where $w_{ij} = \frac{j/n}{\sum_{j=1}^{k_2} j/n}$, $Y_{ij}^{(2)} = (Y_{i+j}^{(1)} - Y_{i-j}^{(1)})/(2j/n)$, k_1 is the same as in (9), and k_2 is a positive integer. Figure 2 displays our second order derivative estimators and empirical second derivatives (16) at interior point for the data from Figure 1, where $k_1 = k_2 = k \in \{6, 9, 12, 25, 35, 60\}$. The performance of the our second derivative estimator and empirical second derivative is parallel to the first derivative's case. Note that the k values used here are larger than the counterparts in the first order derivative estimation.

4.2 Higher Order Derivative Estimation

We generalize the method aforementioned to higher order derivatives $m^{(l)}(x_i)$ $(l > 2)$. The method includes two main steps: the first step is to construct a sequence of symmetric difference quotients in which the derivative is the intercept of the linear regression derived by Taylor expansion, and the second step is to estimate the derivative using the LWLSR.

The construction of a difference sequence is particularly important because it determines the estimation accuracy.

When l is odd, set $d = \frac{l+1}{2}$. We linearly combine $m(x_{i\pm j})$'s subject to

$$\sum_{h=1}^{d}[a_{i,jd+h}m(x_{i+jd+h}) + a_{i,-(jd+h)}m(x_{i-(jd+h)})] = m^{(l)}(x_i) + O\left(\frac{j}{n}\right), \quad 0 \le j \le k,$$

where k is a positive integer. We can derive $2d$ equations through Taylor expansion and solve out the $2d$ unknown parameters. Define

$$Y_{ij}^{(l)} = \sum_{h=1}^{d}[a_{i,jd+h}Y_{i+jd+h} + a_{i,-(jd+h)}Y_{i-(jd+h)}].$$

and consider the linear regression

$$Y_{ij}^{(l)} = m^{(l)}(x_i) + \delta_{ij}, \quad 1 \le j \le k,$$

where $\delta_{ij} = \sum_{h=1}^{d}[a_{i,jd+h}\epsilon_{i+jd+h} + a_{i,-(jd+h)}\epsilon_{i-(jd+h)}] + O(\frac{j}{n})$.

When l is even, set $d = \frac{l}{2}$. We linearly combine $m(x_{i\pm j})$'s subject to

$$b_{i,j}m(x_i)+\sum_{h=1}^{d}[a_{i,jd+h}m(x_{i+jd+h})+a_{i,-(jd+h)}m(x_{i-(jd+h)})] = m^{(l)}(x_i)+O\left(\frac{j}{n}\right), \quad 0 \le j \le k,$$

where k is a positive integer. We can derive $2d+1$ equations through Taylor expansion and solve out the $2d+1$ unknown parameters. Define

$$Y_{ij}^{(l)} = b_{i,j}m(x_i) + \sum_{h=1}^{d}[a_{i,jd+h}Y_{i+jd+h} + a_{i,-(jd+h)}Y_{i-(jd+h)}].$$

and consider the linear regression

$$Y_{ij}^{(l)} = m^{(l)}(x_i) + b_{i,j}\epsilon_i + \delta_{ij}, \quad 1 \le j \le k,$$

where $\delta_{ij} = \sum_{h=1}^{d}[a_{i,jd+h}\epsilon_{i+jd+h} + a_{i,-(jd+h)}\epsilon_{i-(jd+h)}] + O(\frac{j}{n})$.

If k is large enough, it is better to keep the j^2/n^2 term like (7) to reduce the estimation bias. With the regression models defined above, we can obtain the higher order derivative estimators and deduce their asymptotic results by similar arguments as in the previous subsection; the details are omitted here.

5. Simulations

In addition to the simple simulations in the previous sections, we conduct more simulation studies in this section to further evaluate the finite-sample performances of the proposed method and compare it with two other well-known methods. To get more comprehensive comparisons, we use estimation curves and mean absolute errors to assess the performances of different methodologies.

5.1 Finite Sample Results of the First Order Derivative Estimation

We first consider the following two regression functions

$$m(x) = \sin(2\pi x) + \cos(2\pi x) + \log(4/3 + x), \quad x \in [-1, 1], \tag{17}$$

and

$$m(x) = 32e^{-8(1-2x)^2}(1 - 2x), \quad x \in [0, 1]. \tag{18}$$

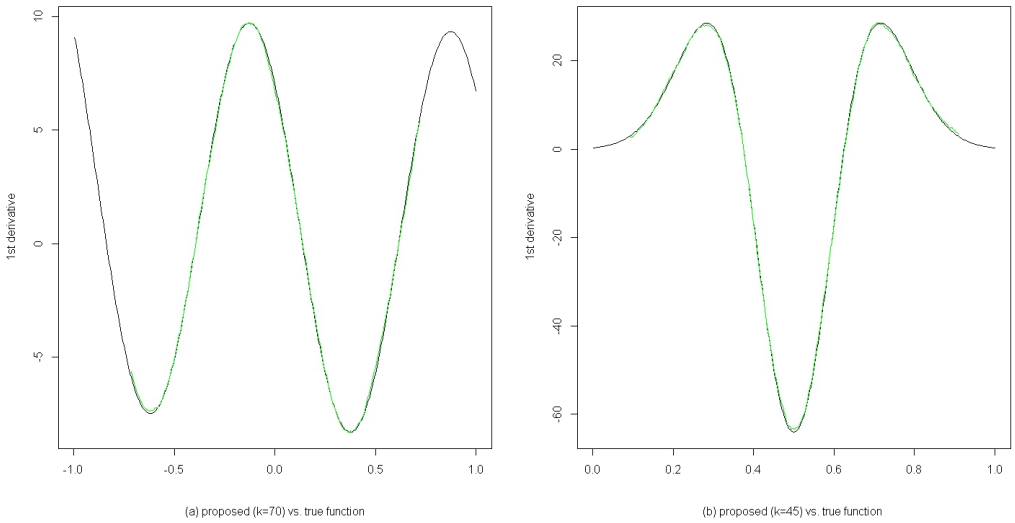


Figure 3: (a) The true first order derivative function (bold line) and our first order derivative estimations (green dashed line). Simulated data set of size 500 from model (1) with equispaced $x_i \in [-1, 1]$, $m(x) = \sin(2\pi x) + \cos(2\pi x) + \log(4/3 + x)$, and $\epsilon_i \overset{iid}{\sim} N(0, 0.1^2)$. (b) The true first order derivative function (bold line) and our first order derivative estimations (green dashed line). Simulated data set of size 500 from model (1) with equispaced $x_i \in [0, 1]$, $m(x) = 32e^{-8(1-2x)^2}(1 - 2x)$, and $\epsilon_i \overset{iid}{\sim} N(0, 0.1^2)$.

These two functions were considered by Hall (2010) and De Brabanter et al. (2013), respectively. The data sets are of size $n = 500$ and generated from model (1) with $\epsilon \sim N(0, \sigma^2)$ for $\sigma = 0.1$. Figure 3 presents the first order derivative estimations of regression functions (17) and (18). It shows that our estimation curves of the first order derivative fit the true curves accurately, although a comparison with the other estimators is not given in the figure.

We now evaluate our estimator with empirical first derivative. Since the oscillation of the periodic function depends on frequency and amplitude, in our simulations we choose the mean function

$$m(x) = A\sin(2\pi f x), \quad x \in [0, 1],$$

			Ours	Empirical	Ours	Empirical	Ours	Empirical
A	f	σ	n=50	n=50	n=200	n=200	n=1000	n=1000
1	1	0.1	0.28(0.09)	0.36(0.08)	0.14(0.04)	0.24(0.04)	0.07(0.02)	0.16(0.03)
		0.5	1.38(0.45)	1.01(0.28)	0.69(0.23)	0.61(0.16)	0.30(0.10)	0.38(0.07)
		2	5.54(1.87)	2.35(0.90)	2.73(0.89)	1.39(0.48)	1.18(0.37)	0.85(0.24)
	2	0.1	0.58(0.15)	1.00(0.13)	0.34(0.07)	0.61(0.07)	0.19(0.03)	0.39(0.04)
		0.5	1.76(0.57)	2.33(0.52)	1.08(0.31)	1.52(0.28)	0.60(0.17)	0.97(0.16)
		2	5.59(1.88)	4.91(1.60)	2.96(1.05)	3.36(0.95)	1.63(0.52)	2.11(0.48)
10	1	0.1	0.41(0.09)	0.98(0.12)	0.24(0.05)	0.67(0.07)	0.13(0.03)	0.42(0.04)
		0.5	1.45(0.47)	2.46(0.44)	0.80(0.22)	1.65(0.25)	0.42(0.10)	1.05(0.14)
		2	5.52(1.76)	5.27(1.24)	2.71(0.89)	3.55(0.76)	1.28(0.35)	2.31(0.38)
	2	0.1	1.15(0.17)	2.90(0.20)	0.64(0.09)	1.66(0.12)	0.35(0.05)	1.06(0.06)
		0.5	3.72(0.79)	6.44(0.77)	2.06(0.39)	4.18(0.42)	1.16(0.19)	2.65(0.24)
		2	9.38(2.78)	13.3(2.40)	5.66(1.38)	9.14(1.35)	3.17(0.72)	5.74(0.65)

Table 1: Adjusted Mean Absolute Error for the first order derivative estimation.

with design points $x_i = i/n$, and the errors are independent and identically normal distribution with zero mean and variance σ^2. We consider three sample sites $n = 50, 200, 1000$, corresponding to small, moderate, and large sample sizes, three standard deviations $\sigma = 0.1, 0.5, 2$, two frequencies $f = 1, 2$, and two amplitudes $A = 1, 10$. The number of repetitions is set as 1000. We consider two criterion: adjusted mean absolute error (AMAE) and mean averaged square error, and find that they have similar performance. For the sake of simplicity and robustness, we choose the AMAE as a measure of comparison. It is defined as

$$AMAE(k) = \frac{1}{n-2k} \sum_{i=k+1}^{n-k} |\hat{m}'(x_i) - m'(x_i)|,$$

here the boundary effects are excluded. According to the condition $k < n/4$ and the AMAE criterion, we choose k as follows

$$\hat{k} = \min\{\arg\min_k AMAE(k), \frac{n}{4}\}.$$

Table 1 reports the simulation results. The numbers outside and inside the brackets are the mean and standard deviation of the AMAE. It indicates our estimator performs better than empirical first derivative in most cases *except that the adoptive k is much less the theoretically uniform k.*

5.2 Finite Sample Results of the Second Order Derivative Estimation

Consider the same functions as in Subsection 5.1. Figure 4 presents the estimation curves and the true curves of the second order derivatives of (17) and (18). It shows that our estimators track the true curves closely.

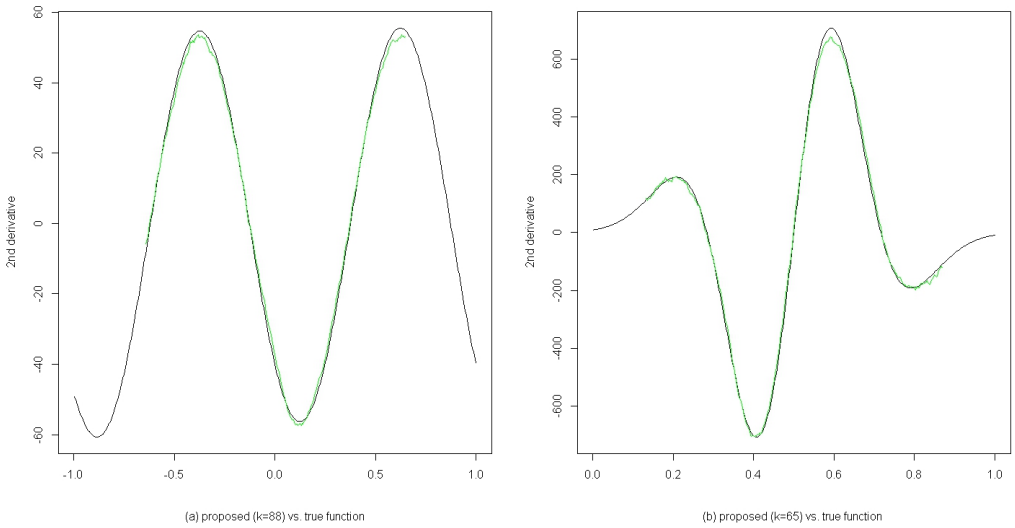


Figure 4: (a)-(b) The true second order derivative function (bold line) and proposed second order derivative estimations (green dashed line) based on the simulated data sets from Figure 3 correspondingly.

	method	n=50	n=100	n=250	n=500
$\sigma = 0.02$	ours	1.03(0.18)	0.79(0.11)	0.62(0.068)	0.47(0.07)
	locpol	1.58(0.45)	0.98(0.22)	0.70(0.11)	0.54(0.08)
	pspline	1.05(0.87)	0.80(0.82)	0.41(0.18)	0.60(0.78)
$\sigma = 0.1$	ours	2.40(0.55)	2.03(0.39)	1.46(0.27)	1.26(0.25)
	locpol	3.90(1.53)	2.93(1.71)	1.79(0.46)	1.52(1.14)
	pspline	2.53(2.32)	3.54(8.33)	1.86(2.79)	2.36(3.91)
$\sigma = 0.5$	ours	6.63(2.05)	5.05(1.61)	4.08(0.96)	3.27(0.90)
	locpol	9.48(2.70)	8.16(5.07)	5.80(3.47)	4.38(2.20)
	pspline	8.23(11.9)	8.00(15.1)	7.52(12.3)	4.77(11.8)

Table 2: Adjusted Mean Absolute Error for the second order derivative estimation.

We evaluate our method with two other well-known methods by Monte Carlo studies, that is local polynomial regression with $p = 5$ (R packages `locpol`, Cabrera, 2012) and penalized smoothing splines with $order = 6$ and $method = 4$ (R packages `pspline`, Ramsay and Ripley, 2013) in model (1). For the sake of simplicity, we set the mean function

$$m(x) = \sin(2\pi x), \quad x \in [-1, 1].$$

We consider four sample sizes, $n \in \{50, 100, 250, 500\}$, and three standard deviations, $\sigma \in \{0.02, 0.1, 0.5\}$. The number of repetitions is set as 100. Table 2 indicates that our estimator is superior to the others in both mean and standard deviation.

6. Discussion

In this paper we propose a new methodology to estimate derivatives in nonparametric regression. The method includes two main steps: construct a sequence of symmetric difference quotients, and estimate the derivative using locally weighted least squares regression. The construction of a difference sequence is particularly important, since it determines the estimation accuracy. We consider three basic principles to construct a difference sequence. First, we eliminate the terms before the derivative of interest through linear combinations, the derivative is thus put in the important place. Second, we adopt every dependent variable only once, which keeps the independence of the difference sequence's terms. Third, we retain one or two terms behind the derivative of interest in the derived linear regression, which reduces estimation bias.

Our method and the local polynomial regression (LPR) have a close relationship. Both methods rely on Taylor expansion and employ the idea of locally weighted fitting. However, there are important differences between them. The first difference is the aim of estimation. The aim of LPR is to estimate the mean, the derivative estimation is only a "by-product", while the aim of our method is to estimate the derivative directly. The second difference is the method of weighting. LPR is kernel-weighted, the farther the distance, the lower the weight; our weight is based on variance, which can be computed exactly. Our simulation studies show that our estimator is more efficient than the LPR in most cases.

All results have been derived for equidistant design with independent identical distributed errors, and extension to more general designs is left to further research. Also, the boundary problem deserve further consideration.

Acknowledgments

We would like to thank Ping Yu at the University of Hong Kong for useful discussions, and two anonymous reviewers and associate editor for their constructive comments on improving the quality of the paper. The research was supported by NNSF projects (11171188, 11571204, and 11231005) of China.

Appendix A. Proof of Theorem 1

For (8), we yield $Var[\hat{\beta}_i] = Var[(D^\top WD)^{-1}D^\top WY_i^{(1)}] = \frac{\sigma^2}{2}(D^\top WD)^{-1}$. We can compute

$$D^\top WD = \begin{pmatrix} I_2/n^2 & I_4/n^4 \\ I_4/n^4 & I_6/n^6 \end{pmatrix},$$

where $I_l = \sum_{j=1}^{k} j^l$, l is an integer. Using the formula for the inverse of a matrix, we have

$$(D^\top WD)^{-1} = \frac{n^8}{I_2I_6 - I_4^2}\begin{pmatrix} I_6/n^6 & -I_4/n^4 \\ -I_4/n^4 & I_2/n^2 \end{pmatrix}.$$

Therefore the variance of $\hat{\beta}_{i0}$ is

$$Var[\hat{\beta}_{i0}] = \frac{\sigma^2}{2}e_1^\top(D^\top WD)^{-1}e_1 = \frac{75\sigma^2}{8}\frac{n^2}{k^3} + o(\frac{n^2}{k^3}).$$

Appendix B. Proof of Theorem 2

From (8), we yield $E[\hat{\beta}_i] = E[(D^\top WD)^{-1}D^\top WY_i^{(1)}] = \beta + (D^\top WD)^{-1}D^\top WE[\delta_i]$. So we have

$$Bias[\hat{\beta}_i] = (D^\top WD)^{-1}D^\top WE[\delta_i].$$

Since m is five times continuously differentiable, the following Taylor expansions are valid for $m(x_{i\pm j})$ around x_i

$$\begin{aligned} m(x_{i\pm j}) = m(x_i) + m^{(1)}(x)(\frac{\pm j}{n}) + \frac{m^{(2)}(x)}{2!}(\frac{\pm j}{n})^2 + \frac{m^{(3)}(x)}{3!}(\frac{\pm j}{n})^3 + \frac{m^{(4)}(x)}{4!}(\frac{\pm j}{n})^4 \\ + \frac{m^{(5)}(x)}{5!}(\frac{\pm j}{n})^5 + o\left((\frac{\pm j}{n})^5\right). \end{aligned}$$

We have

$$\begin{aligned} Y_{ij}^{(1)} &= \frac{Y_{i+j} - Y_{i-j}}{x_{i+j} - x_{i-j}} \\ &= \frac{m^{(1)}(x_i)(\frac{2j}{n}) + \frac{m^{(3)}(x_i)}{3}(\frac{j}{n})^3 + \frac{m^{(5)}(x_i)}{60}(\frac{j}{n})^5 + o\left((\frac{j}{n})^5\right) + (\epsilon_{i+j} - \epsilon_{i-j})}{2j/n} \\ &= m^{(1)}(x_i) + \frac{m^{(3)}(x_i)}{6}(\frac{j}{n})^2 + \frac{m^{(5)}(x_i)}{120}(\frac{j}{n})^4 + o\left((\frac{j}{n})^4\right) + \frac{\epsilon_{i+j} - \epsilon_{i-j}}{2j/n} \\ &= m^{(1)}(x_i) + \frac{m^{(3)}(x_i)}{6}(\frac{j}{n})^2 + \delta_{ij}. \end{aligned}$$

So

$$\begin{aligned} E[\delta_i] &= \frac{m^{(5)}(x_i)}{120}\begin{pmatrix} 1^4/n^4 \\ 2^4/n^4 \\ \vdots \\ k^4/n^4 \end{pmatrix} + o(\begin{pmatrix} 1^4/n^4 \\ 2^4/n^4 \\ \vdots \\ k^4/n^4 \end{pmatrix}), \\ Bias[\hat{\beta}_i] &= \frac{m^{(5)}(x_i)}{120}\frac{k^4}{n^4}\begin{pmatrix} -5/21 \\ 10/9 \end{pmatrix} + o(\frac{k^4}{n^4}). \end{aligned}$$

The estimation bias is $Bias[\hat{\beta}_{i0}] = -\frac{m^{(5)}(x_i)}{504}\frac{k^4}{n^4} + o(\frac{k^4}{n^4})$.

Appendix C. Proof of Theorem 3

Using the asymptotic theory of least squares and the fact that $\{\delta_{ij}\}_{j=1}^k$ are independent distributed with mean zeros and variance $\{\frac{n^2\sigma^2}{2j^2}\}_{j=1}^k$, it follows that the asymptotic normality is proved.

Appendix D. Proof of Corollary 4

For the first derivative estimation, the mean square error is given by

$$\begin{aligned} MSE[\hat{m}^{(1)}(x_i)] &= (Bias[\hat{m}^{(1)}(x_i)])^2 + Var[\hat{m}^{(1)}(x_i)] \\ &= \frac{(m^{(5)}(x_i))^2}{254016}\frac{k^8}{n^8} + \frac{75\sigma^2}{8}\frac{n^2}{k^3} + o(\frac{k^8}{n^8}) + o(\frac{n^2}{k^3}). \end{aligned}$$

Ignoring higher order terms, we obtain the asymptotic mean square error

$$AMSE(\hat{m}^{(1)}(x_i)) = \frac{(m^{(5)}(x_i))^2}{254016}\frac{k^8}{n^8} + \frac{75\sigma^2}{8}\frac{n^2}{k^3}. \tag{19}$$

To minimize (19) with respect to k, we take the first derivative of (19) and yield the gradient

$$\frac{d[AMSE(\hat{m}^{(1)}(x_i))]}{dk} = \frac{(m^{(5)}(x_i))^2}{31752}\frac{k^7}{n^8} - \frac{225\sigma^2}{8}\frac{n^2}{k^4},$$

our optimization problem is to solve $\frac{d[AMSE(\hat{m}^{(1)}(x_i))]}{dk} = 0$. So we obtain

$$k_{opt} = \left(\frac{893025\sigma^2}{(m^{(5)}(x_i))^2}\right)^{1/11} n^{10/11} \doteq 3.48\left(\frac{\sigma^2}{(m^{(5)}(x_i))^2}\right)^{1/11} n^{10/11},$$

and

$$AMSE(\hat{m}^{(1)}(x_i)) \doteq 0.31(\sigma^{16}(m^{(5)}(x_i))^6)^{1/11} n^{-8/11}.$$

Appendix E. Proof of Theorem 5

The conditional variance of $\hat{\beta}_{i0}$ is $Var[\hat{\beta}_{i0}|\epsilon_i] = \sigma^2(D^\top WD)^{-1} = 12\sigma^2\frac{n^2}{k^3} + o(\frac{n^2}{k^3})$.

Since the conditional bias of δ_{ij} is

$$E[\delta_{ij}|\epsilon_i] = \frac{m^{(2)}(x_i)}{2!}\frac{j^1}{n^1} + o\left(\frac{j^1}{n^1}\right).$$

Thus the conditional bias of $\hat{\beta}_{i0}$ is

$$\begin{aligned} Bias[\hat{\beta}_{i0}|\epsilon_i] &= (D^\top WD)^{-1}D^\top WE[\delta_i] \\ &= \frac{m^{(2)}(x_i)}{2}\frac{k^1}{n^1} + o\left(\frac{k^1}{n^1}\right). \end{aligned}$$

Appendix F. Proof of Theorem 6

The conditional variance is given by $Var[\hat{\beta}_i|\epsilon_i] = 2\sigma^2(D^\top WD)^{-1}$. We can compute

$$D^\top WD = \begin{pmatrix} I_4/n^4 & I_6/n^6 & I_2/n^2 \\ I_6/n^6 & I_8/n^8 & I_4/n^4 \\ I_2/n^2 & I_4/n^4 & I_0/n^0 \end{pmatrix}.$$

The determinant of $D^\top WD$ is

$$|D^\top WD| = \frac{I_0I_4I_8 + 2I_2I_4I_6 - I_0I_6^2 - I_4^3 - I_2^2I_8}{n^{12}},$$

and the adjoint matrix is

$$(D^\top WD)^\star = \begin{pmatrix} (I_0I_8 - I_4^2)/n^8 & (I_2I_4 - I_0I_6)/n^6 & (I_4I_6 - I_2I_8)/n^{10} \\ (I_2I_4 - I_0I_6)/n^6 & (I_0I_4 - I_2^2)/n^4 & (I_2I_6 - I_4^2)/n^8 \\ (I_4I_6 - I_2I_8)/n^{10} & (I_2I_6 - I_4^2)/n^8 & (I_4I_8 - I_6^2)/n^{12} \end{pmatrix}.$$

Based on the formula for the inverse of a matrix $A^{-1} = \frac{1}{|A|}A^\star$, we have

$$Var[\hat{\beta}_{i0}|\epsilon_i] = 2\sigma^2 e_1^\top (D^\top W D)^{-1} e_1 = \frac{2205\sigma^2}{8}\frac{n^4}{k^5} + o(\frac{n^4}{k^5}).$$

Revisit the sixth order Taylor approximation for $m(x_{i\pm j})$ around x_i

$$\begin{aligned} m(x_{i\pm j}) = m(x_i) + m^{(1)}(x_i)(\frac{\pm j}{n}) + \frac{m^{(2)}(x_i)}{2!}(\frac{\pm j}{n})^2 + \frac{m^{(3)}(x_i)}{3!}(\frac{\pm j}{n})^3 + \frac{m^{(4)}(x_i)}{4!}(\frac{\pm j}{n})^4 \\ + \frac{m^{(5)}(x_i)}{5!}(\frac{\pm j}{n})^5 + \frac{m^{(6)}(x_i)}{6!}(\frac{\pm j}{n})^6 + o\left((\frac{\pm j}{n})^6\right). \end{aligned}$$

We have

$$\begin{aligned} Y_{ij}^{(2)} &= \frac{m(x_{i-j}) - 2m(x_i) + m(x_{i+j})}{j^2/n^2} + \frac{\epsilon_{i-j} - 2\epsilon_i + \epsilon_{i+j}}{j^2/n^2} \\ &= m^{(2)}(x_i) + \frac{m^{(4)}(x_i)}{12}\frac{j^2}{n^2} + (-2\epsilon_i)\frac{n^2}{j^2} + \frac{m^{(6)}(x_i)}{360}\frac{j^4}{n^4} + o\left(\frac{j^4}{n^4}\right) + \frac{\epsilon_{i-j} + \epsilon_{i+j}}{j^2/n^2}. \end{aligned}$$

So the conditional mean is

$$E[\delta_i|\epsilon_i] = \frac{m^{(6)}(x_i)}{360}\begin{pmatrix} 1^4/n^4 \\ 2^4/n^4 \\ \vdots \\ k^4/n^4 \end{pmatrix} + o(\begin{pmatrix} 1^4/n^4 \\ 2^4/n^4 \\ \vdots \\ k^4/n^4 \end{pmatrix}),$$

and the conditional bias is

$$\begin{aligned} Bias[\hat{\beta}_i|\epsilon_i] &= (D^\top W D)^{-1} D^\top W E[\delta_i|\epsilon_i] \\ &= \frac{m^{(6)}(x_i)}{360}\begin{pmatrix} 5/11 & & \\ & 15/11 & \\ & & 5/231 \end{pmatrix}\begin{pmatrix} k^4/n^4 \\ k^2/n^2 \\ k^6/n^6 \end{pmatrix} + o(\begin{pmatrix} k^4/n^4 \\ k^2/n^2 \\ k^6/n^6 \end{pmatrix}). \end{aligned}$$

We get

$$Bias[\hat{\beta}_{i0}|\epsilon_i] = -\frac{m^{(6)}(x_i)}{792}\frac{k^4}{n^4} + o(\frac{k^4}{n^4}).$$

Appendix G. Proof of Corollary 7

Using the asymptotic theory of least square and the fact that $\{\delta_{ij}\}_{j=1}^k$ are independent distributed with conditional mean zeros and conditional variance $\{\frac{2\sigma^2 n^4}{j^4}\}_{j=1}^k$, it follows that the asymptotic normality is proved.

Appendix H. Proof of Corollary 8

For the second derivative estimation, the MSE is

$$\begin{aligned} MSE[\hat{m}^{(2)}(x_i)|\epsilon_i] &= Bias[\hat{m}^{(2)}(x_i)]^2 + Var[\hat{m}^{(2)}(x_i)] \\ &= \frac{(m^{(6)}(x_i))^2}{627264}\frac{k^8}{n^8} + \frac{2205\sigma^2}{8}\frac{n^4}{k^5} + o(\frac{n^4}{k^5}) + o(\frac{k^8}{n^8}). \end{aligned}$$

Ignoring higher order terms, we get AMSE

$$AMSE[\hat{m}^{(2)}(x_i)|\epsilon_i] = \frac{(m^{(6)}(x_i))^2}{627264}\frac{k^8}{n^8} + \frac{2205\sigma^2}{8}\frac{n^4}{k^5}. \tag{20}$$

To minimize (20) with respect to k, take the first derivative of (20) and yield the gradient

$$\frac{d[AMSE[\hat{m}^{(2)}(x_i)|\epsilon_i]]}{dk} = \frac{(m^{(6)}(x_i))^2}{78408}\frac{k^7}{n^8} - \frac{11025\sigma^2}{8}\frac{n^4}{k^6},$$

our optimization problem is to solve $\frac{d[AMSE[\hat{m}^{(2)}(x_i)|\epsilon_i]]}{dk} = 0$. So we obtain

$$k_{opt} = \left(\frac{108056025\sigma^2}{(m^{(6)}(x_i))^2}\right)^{1/13} n^{12/13} \doteq 4.15\left(\frac{\sigma^2}{(m^{(6)}(x_i))^2}\right)^{1/13} n^{12/13},$$

and

$$AMSE(\hat{m}^{(1)}(x_i)) \doteq 0.36\left(\sigma^{16}(m^{(6)}(x_i))^{10}\right)^{1/13} n^{-8/13}.$$

Appendix I. Convergence Rates

In Table 3, we give the convergence rate of mean estimator and the first order derivative estimator in LPR. $p = 1$ means that the order of LPR is 1. Var_0 represents the convergence rate of the variance of the mean estimator, Var_1 represents the convergence rate of the variance of the first order derivative estimator. $\widetilde{MSE_1}$ stands for the convergence rate of the mean square error of first order derivative estimator when $k = k_0$,

	Var_0	$Bias_0^2$	k_0	MSE_0	Var_1	$Bias_1^2$	k_1	MSE_1	$\widetilde{MSE_1}$
p=1	$1/k$	k^4/n^4	$n^{4/5}$	$n^{-4/5}$	n^2/k^3	k^4/n^4	$n^{6/7}$	$n^{-4/7}$	$n^{-2/5}$
p=2	$1/k$	k^8/n^8	$n^{8/9}$	$n^{-8/9}$	n^2/k^3	k^4/n^4	$n^{6/7}$	$n^{-4/7}$	$n^{-4/9}$
p=3	$1/k$	k^8/n^8	$n^{8/9}$	$n^{-8/9}$	n^2/k^3	k^8/n^8	$n^{10/11}$	$n^{-8/11}$	$n^{-2/3}$
p=4	$1/k$	k^{12}/n^{12}	$n^{12/13}$	$n^{-12/13}$	n^2/k^3	k^8/n^8	$n^{10/11}$	$n^{-8/11}$	$n^{-8/13}$

Table 3: The convergence rates for mean estimator and the first order derivative estimator.

References

J.L.O. Cabrera. locpol: Kernel local polynomial regression. R packages version 0.6-0, 2012. URL `http://mirrors.ustc.edu.cn/CRAN/web/packages/locpol/index.html`.

R. Charnigo, M. Francoeur, M.P. Mengüç, A. Brock, M. Leichter, and C. Srinivasan. Derivatives of scattering profiles: tools for nanoparticle characterization. *Journal of the Optical Society of America*, 24(9):2578–2589, 2007.

R. Charnigo, B. Hall, and C. Srinivasan. A generalized C_p criterion for derivative estimation. *Technometrics*, 53(3):238–253, 2011.

P. Chaudhuri and J.S. Marron. SiZer for exploration of structures in curves. *Journal of the American Statistical Association*, 94(447):807–823, 1999.

K. De Brabanter, J. De Brabanter, B. De Moor, and I. Gijbels. Derivative estimation with local polynomial fitting. *Journal of Machine Learning Research*, 14(1):281–301, 2013.

M. Delecroix and A.C. Rosa. Nonparametric estimation of a regression function and its derivatives under an ergodic hypothesis. *Journal of Nonparametric Statistics*, 6(4):367–382, 2007.

R.L. Eubank and P.L. Speckman. Confidence bands in nonparametric regression. *Journal of the American Statistical Association*, 88(424):1287–1301, 1993.

J. Fan and I. Gijbels. *Local Polynomial Modelling and Its Applications*. Chapman & Hall, London, 1996.

I. Gijbels and A.-C. Goderniaux. Data-driven discontinuity detection in derivatives of a regression function. *Communications in Statistics-Theory and Methods*, 33(4):851–871, 2005.

B. Hall. *Nonparametric Estimation of Derivatives with Applications*. PhD thesis, University of Kentucky, Lexington, Kentucky, 2010.

P. Hall, J.W. Kay, and D.M. Titterington. Asymptotically optimal difference-based estimation of variance in nonparametric regression. *Biometrika*, 77(3):521–528, 1990.

W. Härdle. *Applied Nonparametric Regression*. Cambridge University Press, Cambridge, 1990.

W. Härdle, M. Müller, S. Sperlich, and A. Werwatz. *Nonparametric and Semiparametric Models: An Introduction*. Springer, Berlin, 2004.

N.E. Heckman and J.O. Ramsay. Penalized regression with model-based penalties. *The Canadian Journal of Statistics*, 28(2):241–258, 2000.

J.L. Horowitz. *Semiparametric and Nonparametric Methods in Econometrics*. Springer, New York, 2009.

L. Lin and F. Li. Stable and bias-corrected estimation for nonparametric regression models. *Journal of Nonparametric Statistics*, 20(4):283–303, 2008.

H.-G. Müller. *Nonparametric Regression Analysis of Longitudinal Data*. Springer, New York, 1988.

H.-G. Müller, U. Stadtmüller, and T. Schmitt. Bandwidth choice and confidence intervals for derivatives of noisy data. *Biometrika*, 74(4):743–749, 1987.

J. Newell and J. Einbeck. A comparative study of nonparametric derivative estimators. In *Proceedings of the 22nd International Workshop on Statistical Modelling*, pages 449–452, Barcelona, 2007.

J. Newell, J. Einbeck, N. Madden, and K. McMillan. Model free endurance markers based on the second derivative of blood lactate curves. In *Proceedings of the 20th International Workshop on Statistical Modelling*, pages 357–364, Sydney, 2005.

C. Park and K.-H Kang. SiZer analysis for the comparison of regression curves. *Computational Statistics & Data Analysis*, 52(8):3954–3970, 2008.

J. Ramsay. Derivative estimation. StatLib-S news, 1998. URL `http://www.math.yorku.ca/Who/Faculty/Monette/S-news/0556.html`.

J. Ramsay and B. Ripley. pspline: Penalized smoothing splines. R packages version 1.0-16, 2013. URL `http://mirrors.ustc.edu.cn/CRAN/web/packages/pspline/index.html`.

J.O. Ramsay and B.W. Silverman. *Applied Functional Data Analysis: Methods and Case Studies*. Springer, New York, 2002.

D. Ruppert and M.P. Wand. Multivariate locally weighted least squares regression. *The Annals of Statistics*, 22(3):1346–1370, 1994.

D. Ruppert, S.J. Sheather, and M.P. Wand. An effective bandwidth selector for local least squares regression. *Journal of the American Statistical Association*, 90(432):1257–1270, 1995.

B. Seifert, T. Gasser, and A. Wolf. Nonparametric estimation of residual variance revisited. *Biometrika*, 80(2):373–383, 1993.

C.J. Stone. Additive regression and other nonparametric models. *Annals of Statistics*, 13 (2):689–705, 1985.

T. Tong and Y. Wang. Estimating residual variance in nonparametric regression using least squares. *Biometrika*, 92(4):821–830, 2005.

S. Zhou and D.A. Wolfe. On derivative estimation in spline regression. *Statistica Sinica*, 10 (1):93–108, 2000.

Journal of Machine Learning Research 16 (2015) 2643-2694 Submitted 8/13; Revised 10/14; Published 12/15

When Are Overcomplete Topic Models Identifiable? Uniqueness of Tensor Tucker Decompositions with Structured Sparsity

Animashree Anandkumar A.ANANDKUMAR@UCI.EDU
Department of Electrical Engineering and Computer Science
University of California, Irvine
Engineering Hall, #4408
Irvine, CA 92697, USA

Daniel Hsu DJHSU@CS.COLUMBIA.EDU
Department of Computer Science
Columbia University
1214 Amsterdam Avenue, #0401
New York, NY 10027, USA

Majid Janzamin MJANZAMI@UCI.EDU
Department of Electrical Engineering and Computer Science
University of California, Irvine
Engineering Hall, #4407
Irvine, CA 92697, USA

Sham Kakade SHAM@CS.WASHINGTON.EDU
Department of Computer Science
Department of Statistics
University of Washington
Seattle, WA 98195, USA

Editor: Benjamin Recht

Abstract

Overcomplete latent representations have been very popular for unsupervised feature learning in recent years. In this paper, we specify which overcomplete models can be identified given observable moments of a certain order. We consider probabilistic admixture or topic models in the overcomplete regime, where the number of latent topics can greatly exceed the size of the observed word vocabulary. While general overcomplete topic models are not identifiable, we establish *generic* identifiability under a constraint, referred to as *topic persistence*. Our sufficient conditions for identifiability involve a novel set of "higher order" expansion conditions on the *topic-word matrix* or the *population structure* of the model. This set of higher-order expansion conditions allow for overcomplete models, and require the existence of a perfect matching from latent topics to higher order observed words. We establish that random structured topic models are identifiable w.h.p. in the overcomplete regime. Our identifiability results allows for general (non-degenerate) distributions for modeling the topic proportions, and thus, we can handle arbitrarily correlated topics in our framework. Our identifiability results imply uniqueness of a class of tensor decompositions with structured sparsity which is contained in the class of *Tucker* decompositions, but is more general than the *Candecomp/Parafac* (CP) decomposition.

Keywords: overcomplete representations, topic models, generic identifiability, tensor decomposition

1. Introduction

The performance of many machine learning methods is hugely dependent on the choice of data representations or features. Overcomplete representations, where the number of features can be greater than the dimensionality of the input data, have been extensively employed, and are arguably critical in a number of applications such as speech and computer vision (Bengio et al., 2012). Overcomplete representations are known to be more robust to noise, and can provide greater flexibility in modeling (Lewicki et al., 1998). Unsupervised estimation of overcomplete representations has been hugely popular due to the availability of large-scale unlabeled samples in many applications.

A probabilistic framework for incorporating features posits latent or hidden variables that can provide a good explanation to the observed data. Overcomplete probabilistic models can incorporate a much larger number of latent variables compared to the observed dimensionality. In this paper, we characterize the conditions under which overcomplete latent variable models can be identified from their observed moments.

For any parametric statistical model, identifiability is a fundamental question of whether the model parameters can be uniquely recovered given the observed statistics. Identifiability is crucial in a number of applications where the latent variables are the quantities of interest, e.g. inferring diseases (latent variables) through symptoms (observations), inferring communities (latent variables) via the interactions among the actors in a social network (observations), and so on. Moreover, identifiability can be relevant even in predictive settings, where feature learning is employed for some higher level task such as classification. For instance, non-identifiability can lead to the presence of non-isolated local optima for optimization-based learning methods, and this can affect their convergence properties, e.g., see Uschmajew (2012).

In this paper, we characterize identifiability for a popular class of latent variable models, known as the *admixture* or *topic* models (Blei et al., 2003; Pritchard et al., 2000). These are hierarchical mixture models, which incorporate the presence of multiple latent states (i.e. topics) in each document consisting of a tuple of observed variables (i.e. words). Previous works have established that the model parameters can be estimated efficiently using low order observed moments (second and third order) under some non-degeneracy assumptions, e.g. Anandkumar et al. (2012b); Anandkumar et al. (2012); Arora et al. (2012b). However, these non-degeneracy conditions imply that the model is undercomplete, i.e., the latent dimensionality (number of topics) cannot exceed the observed dimensionality (word vocabulary size). In this paper, we remove this restriction and consider overcomplete topic models, where the number of topics can far exceed the word vocabulary size.

It is perhaps not surprising that general topic models are not identifiable in the overcomplete regime. To this end, we introduce an additional constraint on the model, referred to as *topic persistence*, which roughly means that topics (i.e. latent states) persist locally in a sequence of observed words (but not necessarily globally). This "locality" effect among the observed words is not present in the usual "bag-of-words" or *exchangeable* topic model. Such local dependencies among observations abound in applications such as text, images

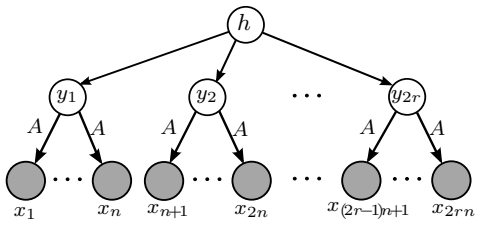


Figure 1: Hierarchical structure of the n-persistent topic model is illustrated for $2rn$ number of words (views) where $r \geq 1$ is an integer. A single topic $y_j, j \in [2r]$, is chosen for each sequence of n views $\{x_{(j-1)n+1}, \ldots, x_{(j-1)n+n}\}$. Matrix A is the population structure or topic-word matrix.

and speech, and can lead to a more faithful representation. In addition, we establish that the presence of topic persistence is central towards obtaining model identifiability in the overcomplete regime, and we provide an in-depth analysis of this phenomenon in this paper.

1.1 Summary of Results

In this paper, we provide conditions for *generic*[1] model identifiability of overcomplete topic models given observable moments of a certain order (i.e., having a certain number of words in each document). We introduce the notion of *topic persistence*, and analyze its effect on identifiability. We establish identifiability in the presence of a novel combinatorial object, referred to as *perfect n-gram matching*, in the bipartite graph from topics to words. Finally, we prove that random structured topic models satisfy these criteria, and are thus identifiable in the overcomplete regime.

1.1.1 Persistent Topic Model

We first introduce the n-persistent topic model, where the parameter n determines the persistence level of a common topic in a sequence of n successive words. For instance, in Figure 1, the sequence of successive words $x_1, \ldots, x_n$ share a common topic y_1, and similarly, the words $x_{n+1}, \ldots, x_{2n}$ share topic y_2, and so on. The n-persistent model reduces to the popular "bag-of-words" model, when $n = 1$, and to the single topic model (i.e. only one topic in each document) when $n \to \infty$. Intuitively, topic persistence aids identifiability since we have multiple *views* of the common hidden topic generating a sequence of successive words. We establish that the bag-of-words model (with $n = 1$) is too non-informative about the topics in the overcomplete regime, and is therefore, not identifiable. On the other hand, n-persistent overcomplete topic models with $n \geq 2$ can become identifiable, and we establish a set of transparent conditions for identifiability.

1. A model is generically identifiable, if all the parameters in the parameter space are identifiable, almost surely. Refer to Definition 2 for more discussion.

1.1.2 Deterministic Conditions for Identifiability

Our sufficient conditions for identifiability are in the form of expansion conditions from the latent topic space to the observed word space. In the overcomplete regime, there are more topics than words in the vocabulary, and thus it is impossible to have expansion on the bipartite graph from topics to words, i.e., the graph encoding the sparsity pattern of the topic-word matrix. Instead, we impose an expansion constraint from topics to "higher order" words, which allows us to incorporate overcomplete models. We establish that this condition translates to the presence of a novel combinatorial object, referred to as the *perfect n-gram matching*, on the topic-word bipartite graph. Intuitively, the perfect n-gram matching condition implies "diversity" among the higher-order word supports for different topics which leads to identifiability. In addition, we present trade-offs among the following quantities: number of topics, size of the word vocabulary, the topic persistence level, the order of the observed moments at hand, the minimum and maximum degrees of any topic in the topic-word bipartite graph, and the *Kruskal rank* (Kruskal, 1976) of the topic-word matrix, under which identifiability holds. To the best of our knowledge, this is the first work to provide conditions for characterizing identifiability of overcomplete topic models with structured sparsity.

As a corollary of our result, we also show that the expansion condition can be removed if the topic-word matrix is full column rank (and therefore undercomplete) and the model is persistent with persistence level at least two.

1.1.3 Identifiability of Random Structured Topic Models

We explicitly characterize the regime of identifiability for the random setting, where each topic i is supported on a random set of d_i words. Therefore, the bipartite graph from topics to words is a random graph with prescribed degrees for topics. For this random model with q topics, p-dimensional word vocabulary, and topic persistence level n, when $q = O(p^n)$ and $\Theta(\log p) \leq d_i \leq \Theta(p^{1/n})$, for all topics i, the topic-word matrix is identifiable from $2n^{\text{th}}$ order observed moments with high probability. Intuitively, the upper bound on the degrees d_i is needed to limit the overlap of word supports among different topics in the overcomplete regime: as the number of topics q increases (i.e., n increases in the above degree bound), the degree needs to be correspondingly smaller to ensure identifiability, and we make this dependence explicit. Intuitively, as the extent of overcompleteness increases, we need sparser connections from topics to words to ensure sufficient diversity in the word supports among different topics. The lower bound on the degrees is required so that there are enough edges in the topic-word bipartite graph so that various topics can be distinguished from one another. Furthermore, we establish that the size condition $q = O(p^n)$ for identifiability is tight.

As in the deterministic case, we also argue the result in the undercomplete setting and show that if $q \leq O(p)$ and $d_i \geq \Omega(\log p)$, then the topic-word matrix is identifiable from $2n^{\text{th}}$ order observed moment with high probability under the persistent model with persistence level n at least equal to two. Here, the upper bound on the degree is relaxed and hence there is no sparsity constraints on the topic-word matrix.

1.1.4 Implications on Uniqueness of Overcomplete Tucker and CP Tensor Decompositions

We establish that identifiability of an overcomplete topic model is equivalent to uniqueness of decomposition of the observed moment tensor (of a certain order). Our identifiability results for persistent topic models imply uniqueness of a structured class of tensor decompositions, which is contained in the class of *Tucker* decompositions, but is more general than the *candecomp/parafac* (CP) decomposition (Kolda and Bader, 2009). This sub-class of Tucker decompositions involves structured sparsity and symmetry constraints on the *core tensor*, and sparsity constraints on the *inverse factors* of the Tucker decomposition. The structural constraints on the Tucker tensor decomposition are related to the topic model as follows: the sparsity and symmetry constraints on the core tensor are related to the persistence property of the topic model, and the sparsity constraints on the inverse factors are equivalent to the sparsity constraints on the topic-word matrix. For n-persistent topic model with $n = 1$ (bag-of-words model), the tensor decomposition is a general Tucker decomposition, where the core tensor is fully dense, while for $n \to \infty$ (single-topic model), the tensor decomposition reduces to a CP decomposition, i.e. the core tensor is a *diagonal tensor*. For a finite persistence level n, in between these two extremes, the core tensor satisfies certain sparsity and symmetry constraints, which becomes crucial towards establishing identifiability in the overcomplete regime.

1.2 Overview of Techniques

We now provide a short overview of the techniques employed in this paper.

Recap of Identifiability Conditions in Under-complete Setting (Expansion Conditions on Topic-Word Matrix): Our approach is based on the recent results of Anandkumar et al. (2012), where conditions for identifiability of topic models are derived, given pairwise observed moments (specifically, co-occurrence of word-pairs in documents). Consider a topic model with q topics and observed word vocabulary of size p. Let $A \in \mathbb{R}^{p \times q}$ denote the topic-word matrix. Expansion conditions are imposed in Anandkumar et al. (2012) on the topic-word bipartite graph which imply that (generically) the sparsest vectors in the column span of A, denoted by $\mathrm{Col}(A)$, are the columns of A themselves. Thus the topic-word matrix A is identifiable from pairwise moments under expansion constraints. However, these expansion conditions constrain the model to be under-complete, i.e., the number of topics $q \leq p$, the size of the word vocabulary. Therefore, the techniques derived in Anandkumar et al. (2012) are not directly applicable here since we consider overcomplete models.

Identifiability in Overcomplete Setting and Why Topic-Persistence Helps: Pairwise moments are thus not sufficient for identifiability of overcomplete models, and the question is whether higher order moments can yield identifiability. We can view the higher order moments as pairwise moments of another equivalent topic model, which enables us to apply the techniques of Anandkumar et al. (2012). The key question is whether we have expansion in the equivalent topic model, which implies identifiability. For a general topic model (without any topic persistence constraints), it can be shown that for identifiability, we require expansion of the n^{th}-order *Kronecker product* of the original topic-word matrix A, denoted by $A^{\otimes n} \in \mathbb{R}^{p^n \times q^n}$, when given access to $(2n)^{\text{th}}$-order moments, for any integer $n \geq 1$. In the overcomplete regime where $q > p$, $A^{\otimes n}$ cannot expand, and therefore, overcomplete

models are not identifiable in general. On the other hand, we show that imposing the constraint of topic persistence can lead to identifiability. For a n-persistent topic model, given $(2n)^{\text{th}}$-order moments, we establish that identifiability occurs when the n^{th}-order *Khatri-Rao product* of A, denoted by $A^{\odot n} \in \mathbb{R}^{p^n \times q}$, expands. Note that the Khatri-Rao product $A^{\odot n}$ is a sub-matrix of the Kronecker product $A^{\otimes n}$, and the Khatri-Rao product $A^{\odot n}$ can expand as long as $q \leq p^n$. Thus, the property of topic persistence is central towards achieving identifiability in the overcomplete regime.

First-Order Approach for Identifiability of Overcomplete Models (Expansion of n-gram Topic-Word Matrix): We refer to $A^{\odot n} \in \mathbb{R}^{p^n \times q}$ as the n-gram topic-word matrix, and intuitively, it relates topics to n-tuple words. Imposing the expansion conditions derived in Anandkumar et al. (2012) on $A^{\odot n}$ implies that (generically) the sparsest vectors in $\text{Col}(A^{\odot n})$, are the columns of $A^{\odot n}$ themselves. Thus, the topic-word matrix A is identifiable from $(2n)^{\text{th}}$-order moments for a n-persistent topic model. We refer to this as the "first-order" approach since we directly impose the expansion conditions of Anandkumar et al. (2012) on $A^{\odot n}$, without exploiting the additional structure present in $A^{\odot n}$.

Why the First-Order Approach is not Enough: Note that $A^{\odot n} \in \mathbb{R}^{p^n \times q}$ matrix relates topics to n-tuples of words. Thus, the entries of $A^{\odot n}$ are highly correlated, even if the original topic-word matrix A is assumed to be randomly generated. It is non-trivial to derive conditions on A, so that $A^{\odot n}$ expands. Moreover, we establish that $A^{\odot n}$ fails to expand on "small" sets, as required in Anandkumar et al. (2012), when the degrees are sufficiently different [2]. Thus, the first-order approach is highly restrictive in the overcomplete setting.

Incorporating Rank Criterion: Note that $A^{\odot n}$ is highly structured: the columns of $A^{\odot n}$ matrix possess a tensor [3] rank of 1, when $n > 1$. This can be incorporated in our identifiability criteria as follows: we provide conditions under which the sparsest vectors in $\text{Col}(A^{\odot n})$, which also possess a tensor rank of 1, are the columns of $A^{\odot n}$ themselves. This implies identifiability of a n-persistent topic model, when given access to $(2n)^{\text{th}}$-order moments. Note that when a small number of columns of $A^{\odot n}$ are combined, the resulting vector cannot possess a tensor rank of 1, and thus, we can rule out that such sparse combinations of columns using the rank criterion. The maximum such number is at least the *Kruskal rank* [4] of A. Thus, sparse combinations of columns of A (up to the Kruskal rank) can be ruled out using the rank criterion, and we require expansion on $A^{\odot n}$ only on large sets of topics (of size larger than the Kruskal rank). This agrees with the intuition that when the topic-word matrix A has a larger Kruskal rank, it should be easier to identify A, since the Kruskal rank is related to the *mutual incoherence* [5] among the columns of A, see Gandy et al. (2011).

2. For $A^{\odot n}$ to expand on a set of size $s \geq 2$, it is necessary that $s \cdot \binom{d_{\min}+n-1}{n} \geq s + \binom{d_{\max}+n-1}{n}$, where $d_{\min}$ and $d_{\max}$ are the minimum and maximum degrees, and n is the extent of overcompleteness: $q = \Theta(p^n)$. When the model is highly overcomplete (large n) and we require small set expansion (small s), the degrees need to be nearly the same. Thus, it is desirable to impose expansion only on large sets, since it allows for more degree diversity.
3. When any column of $A^{\odot n} \in \mathbb{R}^{p^n \times q}$ (of length p^n) is reshaped as a n^{th}-order tensor $T \in \mathbb{R}^{p \times p \times \cdots \times p}$, the tensor T is rank 1.
4. The Kruskal rank is the maximum number k such that every k-subset of columns of A are linearly independent. Note that the Kruskal rank is equal to the rank of A, when A has full column rank. But this cannot happen in the overcomplete setting.
5. It is easy to show that $\text{krank}(A) \geq (\max_{i \neq j} |a_i^\top a_j|)^{-1}$, where a_i, a_j are any pair of columns of A. Thus, higher incoherence leads to a larger kruskal rank.

Notion of Perfect n-gram Matching and Final Identifiability Conditions: Thus, we establish identifiability of overcomplete topic models subject to expansion conditions $A^{\odot n}$ on sets of size larger than the Kruskal rank of the topic-word matrix A. However, it is desirable to impose transparent and interpretable conditions directly on A for identifiability. We introduce the notion of *perfect n-gram matching* on the topic-word bipartite graph, which ensures that each topic can be uniquely matched to a n-tuple word. This combined with a lower bound on the Kruskal rank provides the final set of deterministic conditions for identifiability of the overcomplete topic model. Intuitively, we require that the columns of A be sparse, while still maintaining a large enough Kruskal rank; in other words, the topics have to be sparse and have sufficiently diverse word supports. Thus, we establish identifiability under a set of transparent conditions on the topic-word matrix A, consisting of perfect n-gram matching condition and a lower bound on the Kruskal rank of A.

Analysis under Random-Structured Topic-Word Matrices: Finally, we establish that the derived deterministic conditions are satisfied when the topic-word bipartite graph is randomly generated, as long as the degrees satisfy certain lower and upper bounds. Intuitively, a lower bound on the degrees of the topics is required to have degree concentration on various subsets so that expansion can occur, while the upper bound is required so that the Kruskal rank of the topic-word matrix is large enough compared to the sparsity level. Here, the main technical result is establishing the presence of a perfect n-gram matching in a random bipartite graph with a wide range of degrees. We present a greedy and a recursive mechanism for constructing such a n-gram matching for overcomplete models, which can be relevant even in other settings. For instance, our results imply the presence of a perfect matching when the edges of a bipartite graph are correlated in a structured manner, as given by the Khatri-Rao product.

1.3 Related Works

We now summarize some recent related works in the area of identifiability and learning of latent variable models.

1.3.1 Identifiability, Learning and Applications of Overcomplete Latent Representations

Many recent works employ unsupervised estimation of overcomplete features for higher level tasks such classification, e.g. Coates et al. (2011); Le et al. (2011); Deng and Yu (2013); Bengio et al. (2012), and record huge gains over other approaches in a number of applications such as speech recognition and computer vision. However, theoretical understanding regarding learnability or identifiability of overcomplete representations is far more limited.

Overcomplete latent representations have been analyzed in the context of the independent components analysis (ICA), where the sources are assumed to be independent, and the mixing matrix is unknown. In the overcomplete or under-determined regime of the ICA, there are more sources than sensors. Identifiability and learning of the overcomplete ICA reduces to the problem of finding an overcomplete candecomp/parafac (CP) tensor decomposition. The classical result by Kruskal provides conditions for uniqueness of a CP decomposition (Kruskal, 1976, 1977), with recent extensions to the notion of robust identifiability (Bhaskara et al., 2013). These results provide conditions for strict identifiability

of the model, and here, the dimensionality of the latent space is required to be of the same order as the observed space dimensionality. In contrast, a number of recent works analyze *generic* identifiability of overcomplete CP decomposition, which is weaker than strict identifiability, e.g. Jiang and Sidiropoulos (2004); Lathauwer (2006); Stegeman et al. (June 2006); De Lathauwer et al. (2007); Chiantini and Ottaviani (2012); Bocci et al. (2013); Chiantini et al. (2013). These works assume that the factors (i.e. the components) of the CP decomposition are generically drawn and provide conditions for uniqueness. They allow for the latent dimensionality to be much larger (polynomially larger) than the observed dimensionality. These results on the uniqueness of CP decompositions also lead to identifiability of other latent variable models, such as latent tree models, e.g. Allman et al. (2009, Dec. 2012), and the single-topic model, or more generally latent Dirichlet allocation (LDA). Recently, Goyal et al. (2013) proposed an alternative framework for overcomplete ICA models based on the eigen-decomposition of the reweighted covariance matrix (or higher order moments), where the weights are the Fourier coefficients. However, their approach requires independence of sources (i.e. latent topics in our context), which is not imposed here.

In contrast to the above works dealing with the CP tensor decomposition, we require uniqueness for a more general class of tensor decompositions, in order to establish identifiability of topic models with arbitrarily correlated topics. We establish that our class of tensor decomposition is contained in the class of *Tucker* decompositions which is more general than CP decomposition. Moreover, we explicitly characterize the effect of the sparsity pattern of the factors (i.e., the topic-word matrix) on model identifiability, while all the previous works based on generic identifiability assume fully dense factors (since sparse factors are not generic). For a general overview of tensor decompositions, see Kolda and Bader (2009); Landsberg (2012).

1.3.2 Identifiability and Learning of Undercomplete/Over-determined latent Representations

Much of the theoretical results on identifiability and learning of the latent variable models are limited to non-singular models, which implies that the latent space dimensionality is at most the observed dimensionality. We outline some of the recent works below.

The works of Anandkumar et al. (2012,a,b) provide an efficient moment-based approach for learning topic models, under constraints on the distribution of the topic proportions, e.g. the single topic model, and more generally latent Dirichlet allocation (LDA). In addition, the approach can handle a variety of latent variable models such as Gaussian mixtures, hidden Markov models (HMM) and community models (Anandkumar et al., 2013). The high-level idea is to reduce the problem of learning of the latent variable model to finding a CP decomposition of the (suitably adjusted) observed moment tensor. Various approaches can then be employed to find the CP decomposition. In Anandkumar et al. (2012b), a tensor power method approach is analyzed and is shown to be an efficient guaranteed recovery method in the non-degenerate (i.e. undercomplete) setting. Previously, simultaneous diagonalization techniques have been employed for solving the CP decomposition, e.g. Anandkumar et al. (2012); Mossel and Roch (2006); Chang (1996). However, these techniques fail when the model is overcomplete, as considered here. We note that some recent techniques, e.g. De Lathauwer et al. (2007), can be employed instead, albeit at a cost

of higher computational complexity for overcomplete CP tensor decomposition. However, it is not clear how the sparsity constraints affect the guarantees of such methods. Moreover, these approaches cannot handle general topic models, where the distribution of the topic proportions is not limited to these classes (i.e. either single topic or Dirichlet distribution), and we require tensor decompositions which are more general than the CP decomposition.

There are many other works which consider learning mixture models when multiple views are available. See Anandkumar et al. (2012) for a detailed description of these works. Recently, Rabani et al. (2012) consider learning discrete mixtures given a large number of "views", and they refer to the number of views as the *sampling aperture*. They establish improved recovery results (in terms of ℓ_1 bounds) when sufficient number of views are available ($2k-1$ views for a k-component mixture). However, their results are limited to discrete mixtures or single-topic models, while our setting can handle more general topic models. Moreover, our approach is different since we incorporate sparsity constraints in the topic-word distribution. Another series of recent works by Arora et al. (2012a,b) employ approaches based on non-negative matrix factorization (NMF) to recover the topic-word matrix. These works allow models with arbitrarily correlated topics, as considered here. They establish guaranteed learning when every topic has an *anchor* word, i.e. the word is uniquely generated from that topic, and does not occur under any other topic. Note that the anchor-word assumption cannot be satisfied in the overcomplete setting.

Our work is closely related to the work of Anandkumar et al. (2012) which considers identifiability and learning of topic models under expansion conditions on the topic-word matrix. The work of Spielman et al. (2012b) considers the problem of dictionary learning, which is closely related to the setting of Anandkumar et al. (2012), but in addition assumes that the coefficient matrix is random. However, these works in Anandkumar et al. (2012); Spielman et al. (2012b) can handle only the under-complete setting, where the number of topics is less than the dimensionality of the word vocabulary (or the number of dictionary atoms is less than the number of observations in Spielman et al. (2012b)). We extend these results to the overcomplete setting by proposing novel higher order expansion conditions on the topic-word matrix, and also incorporate additional rank constraints present in higher order moments.

1.3.3 Dictionary Learning/Sparse Coding

Overcomplete representations have been very popular in the context of dictionary learning or sparse coding. Here, the task is to jointly learn a dictionary as well as a sparse selection of the dictionary atoms to fit the observed data. There have been Bayesian as well as frequentist approaches for dictionary learning (Lewicki et al., 1998; Kreutz-Delgado et al., 2003; Rao and Kreutz-Delgado, 1999). However, the heuristics employed in these works (Lewicki et al., 1998; Kreutz-Delgado et al., 2003; Rao and Kreutz-Delgado, 1999) have no performance guarantees. The work of Spielman et al. (2012b) considers learning (undercomplete) dictionaries and provide guaranteed learning under the assumption that the coefficient matrix is random (distributed as Bernoulli-Gaussian variables). Recent works in Mehta and Gray (2013); Maurer et al. (2012) provide generalization bounds for predictive sparse coding, where the goal of the learned representation is to obtain good performance on some predictive task. This differs from our framework since we do not consider predic-

tive tasks here, but the task of recovering the underlying latent representation. Hillar and Sommer (2011) consider the problem of identifiability of sparse coding and establish that when the dictionary succeeds in reconstructing a certain set of sparse vectors, then there exists a unique sparse coding, up to permutation and scaling. However, our setting here is different, since we do not assume that a sparse set of topics occur in each document.

2. Model

We first introduce some notations, and then we provide the persistent topic model.

2.1 Notation

The set $\{1, 2, \ldots, n\}$ is denoted by $[n] := \{1, 2, \ldots, n\}$. Given a set $X = \{1, \ldots, p\}$, set $X^{(n)}$ denotes all ordered n-tuples generated from X. The cardinality of a set S is denoted by $|S|$. For any vector u (or matrix U), the support is denoted by $\mathrm{Supp}(u)$, and the ℓ_0 norm is denoted by $\|u\|_0$, which corresponds to the number of non-zero entries of u, i.e., $\|u\|_0 := |\mathrm{Supp}(u)|$. For a vector $u \in \mathbb{R}^q$, $\mathrm{Diag}(u) \in \mathbb{R}^{q \times q}$ is the diagonal matrix with vector u on its diagonal. The column space of a matrix A is denoted by $\mathrm{Col}(A)$. Vector $e_i \in \mathbb{R}^q$ is the i-th basis vector, with the i-th entry equal to 1 and all the others equal to zero. For $A \in \mathbb{R}^{p \times q}$ and $B \in \mathbb{R}^{m \times n}$, the *Kronecker* product $A \otimes B \in \mathbb{R}^{pm \times qn}$ is defined as (Golub and Loan, 2012)

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \cdots & a_{1q}B \\ a_{21}B & a_{22}B & \cdots & a_{2q}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{p1}B & a_{p2}B & \cdots & a_{pq}B \end{bmatrix},$$

and for $A = [a_1|a_2|\cdots|a_r] \in \mathbb{R}^{p \times r}$ and $B = [b_1|b_2|\cdots|b_r] \in \mathbb{R}^{m \times r}$, the *Khatri-Rao* product $A \odot B \in \mathbb{R}^{pm \times r}$ is defined as

$$A \odot B = [a_1 \otimes b_1 | a_2 \otimes b_2 | \cdots | a_r \otimes b_r].$$

2.2 Persistent Topic Model

In this section, the *n-persistent topic model* is introduced and this imposes an additional constraint, known as topic persistence on the popular admixture model(Blei et al., 2003; Pritchard et al., 2000; Nguyen, 2012). The n-persistent topic model reduces to the bag-of-words admixture model when $n = 1$.

An admixture model specifies a q-dimensional vector of topic proportions $h \in \Delta^{q-1} := \{u \in \mathbb{R}^q : u_i \geq 0, \sum_{i=1}^q u_i = 1\}$ which generates the observed variables $x_l \in \mathbb{R}^p$ through vectors $a_1, \ldots, a_q \in \mathbb{R}^p$. This collection of vectors $a_i, i \in [q]$, is referred to as the *population structure* or the *topic-word matrix* (Nguyen, 2012). For instance, a_i is the conditional distribution of words given topic i. The latent variable h is a q dimensional random vector $h := [h_1, \ldots, h_q]^\top$ known as proportion vector. A prior distribution $P(h)$ over the probability simplex Δ^{q-1} characterizes the prior joint distribution over the latent variables $h_i,\ i \in [q]$. In the topic modeling, this is the prior distribution over the q topics.

The n-persistent topic model has a three-level multi-view hierarchy in Figure 1. $2rn$ number of words (views) are shown in the model for some integer $r \geq 1$. In this model, a common hidden topic is persistent for a sequence of n words $\{x_{(j-1)n+1}, \ldots, x_{(j-1)n+n}\}, j \in [2r]$. Note that the random observed variables (words) are exchangeable within groups of size n, where n is the persistence level, but are not globally exchangeable.

We now describe a linear representation of the n-persistent topic model, on lines of Anandkumar et al. (2012b), but with extensions to incorporate persistence. Each random variable $y_j, j \in [2r]$, is a discrete valued random variable taking one of the q possibilities $\{1, \ldots, q\}$, i.e., $y_j \in [q]$ for $j \in [2r]$. In the n-persistent model, a single common topic is chosen for a sequence of n words $\{x_{(j-1)n+1}, \ldots, x_{(j-1)n+n}\}, j \in [2r]$, i.e., the topic is persistent for n successive views. For notational purposes, we equivalently assume that variables $y_j, j \in [2r]$, are encoded by the basis vectors $e_i,\ i \in [q]$. Thus, the variable $y_j, j \in [2r]$, is

$$y_j = e_i \in \mathbb{R}^q \iff \text{the topic of the } j\text{-th group of words is } i.$$

Given proportion vector h, topics $y_j, j \in [2r]$, are independently drawn according to the conditional expectation

$$\mathbb{E}\big[y_j | h\big] = h, \quad j \in [2r],$$

or equivalently $\Pr\big[y_j = e_i | h\big] = h_i, j \in [2r], i \in [q]$.

Finally, at the bottom layer, each observed variable x_l for $l \in [2rn]$, is a discrete-valued p-dimensional random variable, where p is the size of word vocabulary. Again, we assume that variables x_l, are encoded by the basis vectors $e_k,\ k \in [p]$, such as

$$x_l = e_k \in \mathbb{R}^p \iff \text{the } l\text{-th word in the document is } k.$$

Given the corresponding topic $y_j, j \in [2r]$, words $x_l, l \in [2rn]$, are independently drawn according to the conditional expectation

$$\mathbb{E}\big[x_{(j-1)n+k} | y_j = e_i\big] = a_i,\ i \in [q], j \in [2r],\ k \in [n], \tag{1}$$

where vectors $a_i \in \mathbb{R}^p,\ i \in [q]$, are the conditional probability distribution vectors. The matrix $A = [a_1 | a_2 | \cdots | a_q] \in \mathbb{R}^{p \times q}$ collecting these vectors is the *population structure* or *topic-word matrix*.

The $(2rn)$-th order moment of observed variables $x_l \in \mathbb{R}^p, l \in [2rn]$, for some integer $r \geq 1$, is defined as (in the matrix form) [6]

$$M_{2rn}(x) := \mathbb{E}\left[(x_1 \otimes x_2 \otimes \cdots \otimes x_{rn})(x_{rn+1} \otimes x_{rn+2} \otimes \cdots \otimes x_{2rn})^\top\right] \in \mathbb{R}^{p^{rn} \times p^{rn}}. \tag{2}$$

We now briefly remind why this matrix corresponds to the $(2rn)$-th order moment. Let vectors $\mathbf{i} := (i_1, \ldots, i_{rn})$ and $\mathbf{j} := (j_1, \ldots, j_{rn})$ index the rows and columns of moment matrix $M_{2rn}(x)$. Then, from the above definition, the $(\mathbf{i}, \mathbf{j})$-th entry of $M_{2rn}(x)$ is equal to

$$\mathbb{E}\left[(x_1)_{i_1} \cdots (x_{rn})_{i_{rn}} (x_{rn+1})_{j_1} \cdots (x_{2rn})_{j_{rn}}\right],$$

6. Vector x is the vector generated by concatenating all vectors $x_l, l \in [2rn]$.

which specifies the corresponding $(2rn)$-th observed moment.

For the n-persistent topic model with $2rn$ number of observations (words) $x_l, l \in [2rn]$, the corresponding moment is denoted by $M_{2rn}^{(n)}(x)$. Note that to estimate the $(2rn)^{\text{th}}$ moment, we require a minimum of $2rn$ words in each document. We can select the first $2rn$ words in each document, and average over the different documents to obtain a consistent estimate of the moment. In this paper, we consider the problem of identifiability when exact moments are available.

The moment characterization of the n-persistent topic model is provided in Lemma 2 in Section 4.1. Given $M_{2rn}^{(n)}(x)$, what are the sufficient conditions under which the population structure A is identifiable? This is answered in Section 3.

Remark 1 *Note that our results are valid for the more general linear model $x_l = Ay_j$ (more precisely, $x_{(j-1)n+k} = Ay_j, j \in [2r], k \in [n]$), i.e., each column of matrix A does not need to be a valid probability distribution. Furthermore, the observed random variables x_l, can be continuous while the hidden ones y_j are assumed to be discrete.*

3. Sufficient Conditions for Generic Identifiability

In this section, the identifiability result for the n-persistent topic model with access to $(2n)$-th order observed moment is provided. First, sufficient deterministic conditions on the population structure A are provided for identifiability in Theorem 9. Next, the deterministic analysis is specialized to a random structured model in Theorem 15.

We now make the notion of identifiability precise. As defined in literature, (strict) identifiability means that the population structure A can be uniquely recovered up to permutation and scaling for all $A \in \mathbb{R}^{p \times q}$. Instead, we consider a more relaxed notion of identifiability, known as generic identifiability.

Definition 2 (Generic identifiability) *We refer to a matrix $A \in \mathbb{R}^{p \times q}$ as generic, with a fixed sparsity pattern when the nonzero entries of A are drawn from a distribution which is absolutely continuous with respect to Lebesgue measure*[7]*. For a given sparsity pattern, the class of population structure matrices is said to be* generically identifiable *(Allman et al., Dec. 2012), if all the non-identifiable matrices form a set of Lebesgue measure zero.*

The $(2r)$-th order moment of hidden variables $h \in \mathbb{R}^q$, denoted by $M_{2r}(h) \in \mathbb{R}^{q^r \times q^r}$, is defined as

$$M_{2r}(h) := \mathbb{E}\left[\Big(\overbrace{h \otimes \cdots \otimes h}^{r \text{ terms}}\Big)\Big(\overbrace{h \otimes \cdots \otimes h}^{r \text{ terms}}\Big)^\top\right] \in \mathbb{R}^{q^r \times q^r}. \tag{3}$$

We now provide a set of sufficient conditions for generic identifiability of structured topic models given $(2rn)$-th order observed moment. We first start with a natural assumption on the hidden variables.

Condition 1 (Non-degeneracy) *The $(2r)$-th order moment of hidden variables $h \in \mathbb{R}^q$, defined in equation (3), is full rank (non-degeneracy of hidden nodes).*

7. As an equivalent definition, if the non-zero entries of an arbitrary sparse matrix are independently perturbed with noise drawn from a continuous distribution to generate A, then A is called generic.

Note that there is no hope of distinguishing distinct hidden nodes without this non-degeneracy assumption. We do not impose any other assumption on hidden variables and can incorporate arbitrarily correlated topics.

Furthermore, we can only hope to identify the population structure A up to scaling and permutation. Therefore, we can identify A up to a canonical form defined as:

Definition 3 (Canonical form) *Population structure A is said to be in* canonical form *if all of its columns have unit norm.*

3.1 Deterministic Conditions for Generic Identifiability

In this section, we consider a fixed sparsity pattern on the population structure A and establish generic identifiability when non-zero entries of A are drawn from some continuous distribution. Before providing the main result, a generalized notion of (perfect) matching for bipartite graphs is defined. We subsequently impose these conditions on the bipartite graph from topics to words which encodes the sparsity pattern of population structure A.

3.1.1 Generalized Matching for Bipartite Graphs

A bipartite graph with two disjoint vertex sets Y and X and an edge set E between them is denoted by $G(Y, X; E)$. Given the bi-adjacency matrix A, the notation $G(Y, X; A)$ is also used to denote a bipartite graph. Here, the rows and columns of matrix $A \in \mathbb{R}^{|X|\times|Y|}$ are respectively indexed by X and Y vertex sets. For any subset $S \subseteq Y$, the set of neighbors of vertices in S with respect to A is defined as $N_A(S) := \{i \in X : A_{ij} \neq 0 \text{ for some } j \in S\}$, or equivalently, $N_E(S) := \{i \in X : (j, i) \in E \text{ for some } j \in S\}$ with respect to edge set E.

Here, we define a generalized notion of matching for a bipartite graph and refer to it as n-gram matching.

Definition 4 ((Perfect) n-gram matching) *A* n-gram matching *M for a bipartite graph $G(Y, X; E)$ is a subset of edges $M \subseteq E$ which satisfies the following conditions. First, for any $j \in Y$, we have $|N_M(j)| \leq n$. Second, for any $j_1, j_2 \in Y, j_1 \neq j_2$, we have $\min\{|N_M(j_1)|, |N_M(j_2)|\} > |N_M(j_1) \cap N_M(j_2)|$.*

A perfect n-gram matching *or* Y-saturating n-gram matching *for the bipartite graph $G(Y, X; E)$ is a n-gram matching M in which each vertex in Y is* exactly *connected to n edges in M.*

In words, in a n-gram matching M, each vertex $j \in Y$ is at most connected to n edges in M and for any pair of vertices in Y ($j_1, j_2 \in Y, j_1 \neq j_2$), there exists at least one non-common neighbor in set X for each of them (j_1 and j_2).

As an example, a bipartite graph $G(Y, X; E)$ with $|X| = 4$ and $|Y| = 6$ is shown in Figure 2 for which the edge set E itself is a perfect 2-gram matching.

We also define the following definition of a n-gram matrix.

Definition 5 (n-gram Matrix) *Given a matrix $A \in \mathbb{R}^{p\times q}$, its n-gram matrix $A^{\odot n} \in \mathbb{R}^{p^n\times q}$ is defined as the matrix whose $(\mathbf{i}, j)$-th entry is given by, for $\mathbf{i} := (i_1, i_2, \ldots, i_n) \in [p]^n$ and $j \in [q]$,*

$$A^{\odot n}(\mathbf{i}, j) := A_{i_1,j} A_{i_2,j} \cdots A_{i_n,j}, \quad \text{or} \quad A^{\odot n} := \overbrace{A \odot \cdots \odot A}^{n \text{ times}}.$$

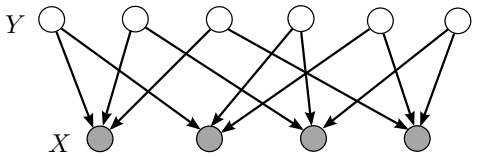


Figure 2: A bipartite graph $G(Y, X; E)$ with $|X| = 4$ and $|Y| = 6$ where the edge set E itself is a perfect 2-gram matching.

That is, $A^{\odot n}$ is the column-wise n^{th} order Kronecker product of n copies of A, and is known as the Khatri-Rao product (Golub and Loan, 2012). Given bipartite graph $G(Y, X; A)$, the notation $G(Y, X^{(n)}; A^{\odot n})$ is also used to denote the bipartite graph corresponding to bi-adjacency matrix $A^{\odot n}$. Here $X^{(n)}$ denotes all ordered n-tuples generated from elements of set X which indexes the rows of $A^{\odot n}$.

The above two definitions might seem unrelated at the first glance, but the following lemma connects them where an interesting property is stated relating the existence of perfect matching in $G(Y, X^{(n)}; A^{\odot n})$ to the existence of perfect n-gram matching in $G(Y, X; A)$. This property is also the original motivation behind defining such notion of generalized matching.

Lemma 1 *If $G(Y, X; A)$ has a perfect n-gram matching, then $G(Y, X^{(n)}; A^{\odot n})$ has a perfect matching. In the other direction, if $G(Y, X^{(n)}; A^{\odot n})$ has a perfect matching $M^{\odot n}$, then $G(Y, X; A)$ has a perfect n-gram matching under the following condition on $M^{\odot n}$. All the matching edges $(j, (i_1, \ldots, i_n)) \in M^{\odot n}$ should satisfy $i_1 \neq i_2 \neq \cdots \neq i_n$ for all $j \in Y$. In words, the matching edges should be connected to nodes in $X^{(n)}$, which are indexed by tuples of distinct indices.*

See Appendix A.4 for the proof.

We also provide more discussions and remarks on the n-gram matching as follows.

Remark 6 (Relationship to other matchings) *The relationship of n-gram matching to other types of matchings is discussed below.*

- *Regular matching: For special case $n = 1$, the (perfect) n-gram matching reduces to the usual (perfect) matching for bipartite graphs.*
- *b-matching: For a bipartite graph $G(Y, X; E)$, a b-matching for vertices in Y is a subset of edges $M_b \subseteq E$, where each vertex in Y is connected to b edges. Comparing with the proposed perfect (Y-saturating) b-gram matching, b-matching does not enforce that the set of neighbors be different.*

Remark 7 (Necessary size bound) *Consider a bipartite graph $G(Y, X; E)$ with $|Y| = q$ and $|X| = p$ which has a perfect n-gram matching. Note that there are $\binom{p}{n}$ n-combinations on X side and each combination can at most have one neighbor (a node in Y which is connected to all nodes in the combination) through the matching, and therefore we necessarily have $q \leq \binom{p}{n}$.*

Finally, note that the existence of perfect n-gram matching results in the existence of perfect $(n+1)$-gram matching [8], but the reverse is not true. For example, the bipartite graph $G(Y, X; E)$ with $|X| = 4$ and $|Y| = \binom{4}{2} = 6$ in Figure 2, has a perfect 2-gram matching, but not a perfect (1-gram) matching (since $6 > 4$).

3.1.2 Identifiability Conditions Based on Existence of Perfect n-gram Matching in Topic-word Graph

Now, we are ready to propose the identifiability conditions and result.

Condition 2 (Perfect n-gram matching on A) *The bipartite graph $G(V_h, V_o; A)$ between hidden and observed variables, has a perfect n-gram matching* [9].

The above condition implies that the sparsity pattern of matrix A is appropriately scattered in the mapping from hidden to observed variables to be identifiable. Intuitively, it means that every hidden node can be distinguished from another hidden node by its unique set of neighbors under the corresponding n-gram matching.

Furthermore, condition 2 is the key to be able to propose identifiability in the overcomplete regime. As stated in the size bound in Remark 7, for $n \geq 2$, the number of hidden variables can be more than the number of observed variables and we can still have perfect n-gram matching.

Definition 8 (Kruskal rank, (Kruskal, 1977)) *The* Kruskal rank *or the* krank *of matrix A is defined as the maximum number k such that every subset of k columns of A is linearly independent.*

Note that krank is different from the general notion of matrix rank and it is a lower bound for the matrix rank, i.e., $\text{Rank}(A) \geq \text{krank}(A)$.

Condition 3 (Krank condition on A) *The Kruskal rank of matrix A satisfies the bound $\text{krank}(A) \geq d_{\max}(A)^n$, where $d_{\max}(A)$ is the maximum node degree of any column of A, i.e., $d_{\max}(A) := \max_{i \in [q]} \|Ae_i\|_0$. Here n is the same as parameter n in Condition 2.*

In the overcomplete regime, it is not possible for A to be full column rank and $\text{krank}(A) < |V_h| = q$. However, note that a large enough krank ensures that appropriate sized subsets of columns of A are linearly independent. For instance, when $\text{krank}(A) > 1$, any two columns cannot be collinear and the above condition rules out the collinear case for identifiability. In the above condition, we see that a larger krank can incorporate denser connections between topics and words.

On the other hand, the bound in Condition 3 imposes sparsity on the columns of topic-word matrix as $d_{\max}(A) \leq \text{krank}(A)^{1/n}$. Under such sparsity constraint, each topic (index-

8. Note that the degree of each node (on matching side Y) in the original bipartite graph should be at least $n + 1$.
9. Parameter n in all of the conditions refer to the same parameter n as the persistence level of the model. Note that we are considering the n-persistent topic model proposed in Section 2.

ing the columns of A) is supported on a specific set of words which enables us to distinguish between different topics and identify the model. But, it seems that this bound is not tight[10].

The main identifiability result under a fixed graph structure is stated in the following theorem for $n \geq 2$, where n is the topic persistence level. The identifiability result relies on having access to the $(2rn)$-th order moment of observed variables $x_l, l \in [2rn]$, defined in equation (2) as

$$M_{2rn}(x) := \mathbb{E}\left[(x_1 \otimes x_2 \otimes \cdots \otimes x_{rn})(x_{rn+1} \otimes x_{rn+2} \otimes \cdots \otimes x_{2rn})^\top\right] \in \mathbb{R}^{p^{rn} \times p^{rn}},$$

for some integer $r \geq 1$.

Theorem 9 (Generic identifiability under deterministic topic-word graph structure) *Let $M_{2rn}^{(n)}(x)$ in equation (2) be the $(2rn)$-th order observed moment of the n-persistent topic model for some integer $r \geq 1$. If the model satisfies conditions 1, 2 and 3, then, for any $n \geq 2$, all the columns of population structure A are* generically identifiable *from $M_{2rn}^{(n)}(x)$. Furthermore, the $(2r)$-th order moment of the hidden variables, denoted by $M_{2r}(h)$, is also* generically identifiable*.*

The theorem is proved in Appendix A. It is seen that the population structure A is identifiable, given any observed moment of order at least $2n$. Increasing the order of observed moment results in identifying higher order moments of the hidden variables.

The above theorem does not cover the case when the persistence level $n = 1$. This is the usual bag-of-words admixture model. Identifiability of this model has been studied earlier in Anandkumar et al. (2012) and we recall it below.

Remark 10 (Bag-of-words admixture model, (Anandkumar et al., 2012)) *Given $(2r)$-th order observed moments with $r \geq 1$, the structure of the popular bag-of-words admixture model and the $(2r)$-th order moment of hidden variables are identifiable, when A is full column rank and the following expansion condition holds (Anandkumar et al., 2012)*

$$|N_A(S)| \geq |S| + d_{\max}(A), \quad \forall S \subseteq V_h,\ |S| \geq 2. \tag{4}$$

Our result for $n \geq 2$ in Theorem 9, provides identifiability in the overcomplete regime with weaker matching condition 2 and krank condition 3. The matching condition 2 is weaker than the above expansion condition which is based on the perfect matching and hence, does not allow overcomplete models. Furthermore, the above result for the bag-of-words admixture model requires full column rank of A which is more stringent than our krank condition 3.

Remark 11 (Kruskal rank and degree diversity) *Condition 3 requires that the Kruskal rank of the topic-word matrix be large enough compared to the maximum degree of the topics. Intuitively, a larger Kruskal rank ensures enough diversity in the word supports among different topics under a higher level of sparsity. This Kruskal rank condition also allows for more degree diversity among the topics, when the topic persistence level $n > 1$. On*

10. The looseness originates from bound (37) as $\left|N_{A_{\text{Rest.}}^{\odot n}}(S)\right| \geq |N_A(S)| + |S|$ in the proof. See Definitions 5 and 25 for the definition of $A_{\text{Rest.}}^{\odot n}$. Note that many terms in this lower bound on $\left|N_{A_{\text{Rest.}}^{\odot n}}(S)\right|$ are ignored which leads to a loose bound that might be improved.

the other hand, for the bag-of-words model ($n = 1$), using (4) implies that $2d_{\min} > d_{\max}$, where $d_{\min}, d_{\max}$ are the minimum and maximum degrees of the topics. Thus, we provide identifiability results with more degree diversity when higher order moments are employed.

Remark 12 (Recovery using ℓ_1 optimization) *It turns out that our conditions for identifiability imply that the columns of the n-gram matrix $A^{\odot n}$, defined in Definition 5, are the sparsest vectors in $\mathrm{Col}\Big(M_{2n}^{(n)}(x)\Big)$, having a tensor rank of one. See Appendix A. This implies recovery of the columns of A through exhaustive search, which is not efficient. On the other hand, efficient ℓ_1-based recovery algorithms have been analyzed in Spielman et al. (2012a); Anandkumar et al. (2012) for the undercomplete case ($n = 1$). They can be employed here for recovery from higher order moments as well. Exploiting additional structure present in $A^{\odot n}$, for $n > 1$, such as rank-1 test devices proposed in De Lathauwer et al. (2007) are interesting avenues for future investigation.*

In Theorem 9, we provide our identifiability result for the overcomplete topic-word matrix A under topic persistent model. The result for the bag-of-words admixture model is also reviewed in Remark 10 under the assumption that A is full column rank. In the following corollary, we provide the strong identifiability result for the full column rank topic-word matrix under the topic persistent model.

Corollary 13 (Identifiability for undercomplete topic-word matrix) *Let $M_{2rn}^{(n)}(x)$ in equation (2) be the $(2rn)$-th order observed moment of the n-persistent topic model for some integer $r \geq 1$. If the model satisfies condition 1, and in addition A is full column rank, then for any $n \geq 2$, all the columns of population structure A are* generically identifiable *from $M_{2rn}^{(n)}(x)$. Furthermore, the $(2r)$-th order moment of the hidden variables, denoted by $M_{2r}(h)$, is also* generically identifiable*.*

Comparing to Theorem 9 and Remark 10, the expansion (and krank) conditions are not required in the above result which is a huge relaxation. The reason is both undercomplete regime and topic persistence are assumed here which relaxes the other conditions. Note that the assumptions that topic persists with persistence $n \geq 2$, and the topic-word matrix is full column rank (and therefore undercomplete) is reasonable in many applications.

3.2 Analysis Under Random Topic-word Graph Structures

In this section, we specialize the identifiability result to the random case. This result is based on more transparent conditions on the size and the degree of the random bipartite graph $G(V_h, V_o; A)$. We consider the random model where in the bipartite graph $G(V_h, V_o; A)$, each node $i \in V_h$ is randomly connected to d_i different nodes in set V_o. Note that this is a heterogeneous degree model.

Furthermore, the random identifiability result is provided with high probability which is defined as follows.

Definition 14 (whp) *A sequence of events $\mathcal{E}_p$ (depending on size parameter p) occurs* with high probability *(**whp**) if $\Pr(\mathcal{E}_p) = 1 - O(p^{-\epsilon})$ for some $\epsilon > 0$.*

Condition 4 (Size condition) *The random bipartite graph $G(V_h, V_o; A)$ with $|V_h| = q, |V_o| = p$, and $A \in \mathbb{R}^{p \times q}$, satisfies the size condition $q \leq \left(c\frac{p}{n}\right)^n$ for some constant $0 < c < 1$.*

Parameter	Representing
p	dimension of observed variables
q	dimension of hidden variables
n	persistence level
c	size ratio such that $q \leq \left(c\frac{p}{n}\right)^n$
α, β	Constants for lower bound on degree such that $d_{\min} \geq \max\{1+\beta\log p, \alpha\log p\}$

Table 1: Table of parameters.

This size condition is required to establish that the random bipartite graph has a perfect n-gram matching (and hence satisfies deterministic condition 2). It is shown in Section 5.2.1 that the necessary size constraint $q = O(p^n)$ stated in Remark 7, is achieved in the random case. Thus, the above constraint allows for the overcomplete regime, where $q \gg p$ for $n \geq 2$, and is tight.

Condition 5 (Degree condition) *In the random bipartite graph $G(V_h, V_o; A)$ with $|V_h| = q, |V_o| = p$, and $A \in \mathbb{R}^{p\times q}$, the degree d_i of nodes $i \in V_h$ satisfies the following lower and upper bounds ($d_i \in [d_{\min}, d_{\max}]$):*

- Lower bound: $d_{\min} \geq \max\{1+\beta\log p, \alpha\log p\}$ *for some constants* $\beta > \frac{n-1}{\log 1/c}, \alpha > \max\left\{2n^2\left(\beta\log\frac{1}{c}+1\right), 2\beta n\right\}$.
- Upper bound: $d_{\max} \leq (cp)^{\frac{1}{n}}$.

Intuitively, the lower bound on the degree is required to show that the corresponding bipartite graph $G(V_h, V_o; A)$ has sufficient number of random edges to ensure that it has perfect n-gram matching with high probability. The upper bound on the degree is mainly required to satisfy the krank condition 3, where $d_{\max}(A)^n \leq \text{krank}(A)$. As discussed after Condition 3, this upper bound is not tight.

It is important to see that, for $n \geq 2$, the above condition on degree covers a range of models from sparse to intermediate regimes and it is reasonable in a number of applications that each topic does not generate a very large number of words.

The proposed parameters in Conditions 4 and 5 are summarized in Table 1.

The main random identifiability result is stated in the following theorem for $n \geq 2$, while $n = 1$ case is addressed in Remark 17. The identifiability result relies on having access to the $(2rn)$-th order moment of observed variables $x_l, l \in [2rn]$, defined in equation (2) as

$$M_{2rn}(x) := \mathbb{E}\left[(x_1 \otimes x_2 \otimes \cdots \otimes x_{rn})(x_{rn+1} \otimes x_{rn+2} \otimes \cdots \otimes x_{2rn})^\top\right] \in \mathbb{R}^{p^{rn}\times p^{rn}},$$

for some integer $r \geq 1$.

Probability rate constants: The probability rate of success in the following random identifiability result is specified by constants $\beta' > 0$ and $\gamma = \gamma_1 + \gamma_2 > 0$ as

$$\beta' = -\beta\log c - n + 1, \tag{5}$$

$$\gamma_1 = e^{n-1}\Big(\frac{c}{n^{n-1}} + \frac{e^2}{1-\delta_1} n^{\beta'+1}\Big), \tag{6}$$

$$\gamma_2 = \frac{c^{n-1}e^2}{n^n(1-\delta_2)}, \tag{7}$$

where δ_1 and δ_2 are some constants satisfying $e^2\Big(\frac{p}{n}\Big)^{-\beta \log 1/c} < \delta_1 < 1$ and $\frac{c^{n-1}e^2}{n^n} p^{-\beta'} < \delta_2 < 1$.

Theorem 15 (Random identifiability) *Let $M_{2rn}^{(n)}(x)$ in equation (2) be the $(2rn)$-th order observed moment of the n-persistent topic model for some integer $r \geq 1$. If the model with random population structure A satisfies conditions 1, 4 and 5, then **whp** (with probability at least $1 - \gamma p^{-\beta'}$ for constants $\beta' > 0$ and $\gamma > 0$, specified in (5)-(7)), for any $n \geq 2$, all the columns of population structure A are identifiable from $M_{2rn}^{(n)}(x)$. Furthermore, the $(2r)$-th order moment of hidden variables, denoted by $M_{2r}(h)$, is also identifiable, **whp**.*

The theorem is proved in Appendix B. Similar to the deterministic analysis, it is seen that the population structure A is identifiable given any observed moment with order at least $2n$. Increasing the order of observed moment results in identifying higher order moments of the hidden variables.

Remark 16 (Trade-off between topic-word size ratio and degree) *When the number of hidden variables increases, i.e. c increases, but the order n is kept fixed, the bounds on degree in condition 5 also needs to grow. Intuitively, a larger degree is needed to provide more flexibility in choosing the subsets of neighbors for hidden nodes to ensure the existence of a perfect n-gram matching in the bipartite graph, which in turn ensures identifiability. Note that as c grows, the parameter β, which is the lower bound on d also grows, and the probability rate (i.e., the term $-\beta \log c$) remains constant. Hence, the probability rate does not change as c increases, since the increase in the degree d compensates the additional "difficulty" arising due to a larger number of hidden variables.*

The above identifiability theorem only covers for $n \geq 2$ and the $n = 1$ case is addressed in the following remark.

Remark 17 (Bag-of-words admixture model) *The identifiability result for the random bag-of-words admixture model is comparable to the result in Spielman et al. (2012a), which considers exact recovery of sparsely-used dictionaries. They assume that $Y = DX$ is given for some unknown arbitrary dictionary $D \in \mathbb{R}^{q\times q}$ and unknown random sparse coefficient matrix $X \in \mathbb{R}^{q\times p}$. They establish that if $D \in \mathbb{R}^{q\times q}$ is full rank and the random sparse coefficient matrix $X \in \mathbb{R}^{q\times p}$ follows the Bernoulli-subgaussian model with size constraint $p > Cq \log q$ and degree constraint $O(\log q) < \mathbb{E}[d] < O(q \log q)$, then the model is identifiable, whp. Comparing the size and degree constraints, our identifiability result for $n \geq 2$ requires more stringent upper bound on the degree ($d = O(p^{1/n})$), while more relaxed condition on the size ($q = O(p^n)$) which allows to identifiability in the overcomplete regime.*

Remark 18 (The size condition is tight) *The size bound $q = O(p^n)$ in the above theorem achieves the necessary condition that $q \leq \binom{p}{n} = O(p^n)$ (see Remark 7), and is therefore tight. The sufficiency is argued in Theorem 22, where we show that the matching condition 2 holds under the above size and degree conditions 4 and 5.*

As in the deterministic case, we finish this section by providing random identifiability result for the full column rank topic-word matrix under the topic persistent model.

Corollary 19 (Random identifiability for undercomplete topic-word matrix) *Let $M_{2rn}^{(n)}(x)$ in equation (2) be the $(2rn)$-th order observed moment of the n-persistent topic model for some integer $r \geq 1$. If the model with random population structure $A \in \mathbb{R}^{p \times q}$ satisfies condition 1, size condition $q \leq cp$ for some constant $0 < c < 1$ and the degree condition $d_{\min} \geq 1 + \beta \log p$ for some constant $\beta > 0$, then **whp** (with probability at least $1 - O(z^{-\beta \log 1/c})$ where $\beta \log \frac{1}{c} > 0$), for any $n \geq 2$, all the columns of population structure A are identifiable from $M_{2rn}^{(n)}(x)$. Furthermore, the $(2r)$-th order moment of hidden variables, denoted by $M_{2r}(h)$, is also identifiable, **whp**.*

Comparing to Theorem 15, the upper bound on the degree (sparsity constraint) is not required in the above result which is a huge relaxation.

4. Identifiability via Uniqueness of Tensor Decompositions

In this section, we characterize the moments of the n-persistent topic model in terms of the model parameters, i.e. the topic-word matrix A and the moment of hidden variables. We relate identifiability of the topic model to uniqueness of a certain class of tensor decompositions, which in turn, enables us to prove Theorems 9 and 15. We then discuss the special cases of the persistent topic model, viz., the single topic model (infinite-persistent topic model) and the bag-of-words admixture model (1-persistent topic model).

4.1 Moment Characterization of the Persistent Topic Model

In the following lemma, which is proved in Appendix A.2, we characterize the observed moments of a persistent topic model. Throughout this section, the order of the observed moment is fixed to $2m$.

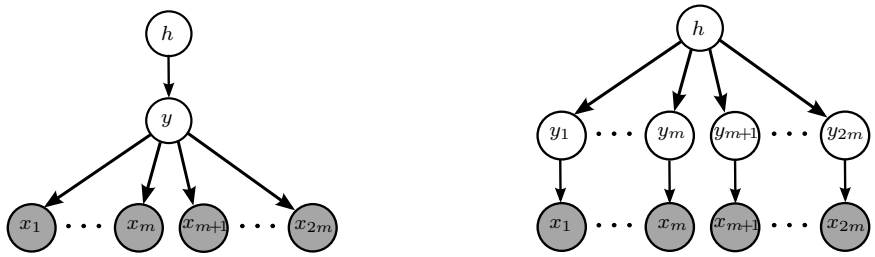
Lemma 2 (n-persistent topic model moment characterization) *The $(2m)$-th order moment of observed variables, defined in equation (2), for the n-persistent topic model is characterized as*[11]*:*

- *if $m = rn$, for some integer $r \geq 1$, then*

$$M_{2m}^{(n)}(x) = \left(\overbrace{A^{\odot n} \otimes \cdots \otimes A^{\odot n}}^{r \text{ times}} \right) M_{2r}(h) \left(\overbrace{A^{\odot n} \otimes \cdots \otimes A^{\odot n}}^{r \text{ times}} \right)^{\top}, \tag{8}$$

  where $M_{2r}(h) \in \mathbb{R}^{q^r \times q^r}$ is the $(2r)$-th order moment of hidden variables $h \in \mathbb{R}^q$, defined in equation (3), and the n-gram matrix $A^{\odot n}$ is defined in Definition 5.

11. The other cases not covered in Lemma 2 are deferred to Appendix A.2. See Remark 30.



(a) Single topic model (infinite-persistent topic model)

(b) Bag-of-words admixture model (1-persistent topic model)

Figure 3: Hierarchical structure of the single topic model and bag-of-words admixture model shown for $2m$ number of words (views).

- *If $n \geq 2m$, then*

$$M_{2m}^{(n)}(x) = \left(A^{\odot m}\right) M_1(h) \left(A^{\odot m}\right)^\top, \tag{9}$$

where $M_1(h) := \mathrm{Diag}(\mathbb{E}[h]) \in \mathbb{R}^{q \times q}$ is the first order moment of hidden variables $h \in \mathbb{R}^q$, stacked in a diagonal matrix.

Thus, we see that the observed moments can be expressed in terms of the hidden moments $M(h)$ and the Kronecker products of the n-gram matrices. In the special case, when the persistence level is large enough compared to the order of the moment ($n \geq 2m$), the moment form reduces to a Khatri-Rao product form in (9). Moreover, in (9), we have a diagonal matrix $M_1(h)$ instead of a general (dense) matrix $M_{2r}(h)$ in (8), when $n < 2m = 2rn$. Thus, we have a more succinct representation of the moments in (9) when the persistence level of the topics is large enough.

In the following, we contrast the special cases when the persistence level n is $n \to \infty$ (single topic model) and $n = 1$ (bag of words admixture model), as shown in Fig.3a and Fig.3b. In order to have a fair comparison, the number of observed variables is fixed to $2m$ and the persistence level is varied.

Single topic model ($n \to \infty$): The condition in (9) ($n \geq 2m$) is always satisfied for the single-topic model, since $n \to \infty$ in this case, and we have

$$M_{2m}^{(\infty)}(x) = \left(A^{\odot m}\right) M_1(h) \left(A^{\odot m}\right)^\top. \tag{10}$$

Note that $M_1(h)$ is a diagonal matrix.

Bag-of-words admixture model ($n = 1$): From Lemma 2, the $(2m)$-th order moment of observed variables $x_l, l \in [2m]$, for the bag-of-words admixture model (1-persistent topic model), shown in Figure 3b, is given by

$$M_{2m}^{(1)}(x) = \Big(\overbrace{A \otimes \cdots \otimes A}^{m \text{ times}}\Big) M_{2m}(h) \Big(\overbrace{A \otimes \cdots \otimes A}^{m \text{ times}}\Big)^\top, \tag{11}$$

where $M_{2m}(h) \in \mathbb{R}^{q^m \times q^m}$ is the $(2m)$-th order moment of hidden variables $h \in \mathbb{R}^q$, defined in (3). Note that $M_{2m}(h)$ is a full matrix in general.

Contrasting single topic ($n \to \infty$) and bag of words models ($n = 1$): Comparing equations (10) and (11), it is seen that the moments under the single topic model in (10) are more "structured" compared to the bag of words model in (11). In (11), we have Kronecker products of the topic-word matrix A, while (10) involves Khatri-Rao products of A. This forms a crucial criterion in determining of whether overcomplete models are identifiable, as discussed below.

Why does persistence help in identifiability of overcomplete models? For simplicity, let the order of the moment $2m = 4$. The equations (10) and (11) reduce to

$$M_4^{(\infty)}(x) = (A \odot A)\,\mathrm{Diag}\left(\mathbb{E}\big[h\big]\right)(A \odot A)^\top, \tag{12}$$

$$M_4^{(1)}(x) = (A \otimes A)\mathbb{E}\big[(h \otimes h)(h \otimes h)^\top\big](A \otimes A)^\top. \tag{13}$$

Note that for the single topic model in (12), the Khatri-Rao product matrix $A \odot A \in \mathbb{R}^{p^2 \times q}$ has the same as the number of columns (i.e. the latent dimensionality) of the original matrix A, while the number of rows (i.e. the observed dimensionality) is increased. Thus, the Khatri-Rao product "expands" the effect of hidden variables to higher order observed variables, which is the key towards identifying overcomplete models. In other words, the original overcomplete representation becomes determined due to the 'expansion effect' of the Khatri-Rao product structure of the higher order observed moments.

On the other hand, in the bag-of-words admixture model in (13), this interesting 'expansion property' does not occur, and we have the Kronecker product $A \otimes A \in \mathbb{R}^{p^2 \times q^2}$, in place of the Khatri-Rao products. The Kronecker product operation increases both the number of the columns (i.e. latent dimensionality) and the number of rows (i.e. observed dimensionality), which implies that higher order moments do not help in identifying overcomplete models.

An example is provided in Figure 4 which helps to see how the matrices $A \odot A$ and $A \otimes A$ behave differently in terms of mapping topics to word tuples.

Note that for the n-persistent model, for $n = 2$, the 4^{th} order moment reduces to

$$M_4^{(2)}(x) = (A \odot A)\mathbb{E}\big[hh^\top\big](A \odot A)^\top. \tag{14}$$

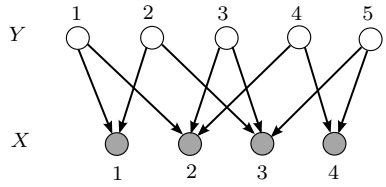
Contrasting the above equation with (12) and (13), we find that the 2-persistent model retains the desirable property of possessing Khatri-Rao products, while being more general than the form for single topic model in (12). This key property enables us to establish identifiability of topic models with finite persistence levels.

4.2 Tensor Algebra of the Moments

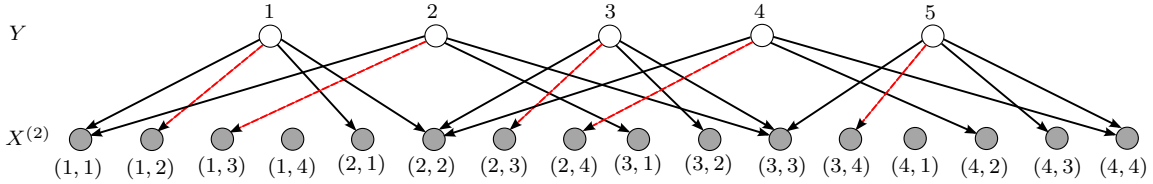
In Section 4.1, we provided a representation of the moment forms in the matrix form. We now provide the equivalent tensor representation of the moments. The tensor representation is more compact and transparent, and allows us to compare the topic models under different levels of persistence. We compare the derived tensor form with the well-known Tucker and CP decompositions. We first introduce some tensor notations and definitions.

4.2.1 Tensor Notations and Definitions

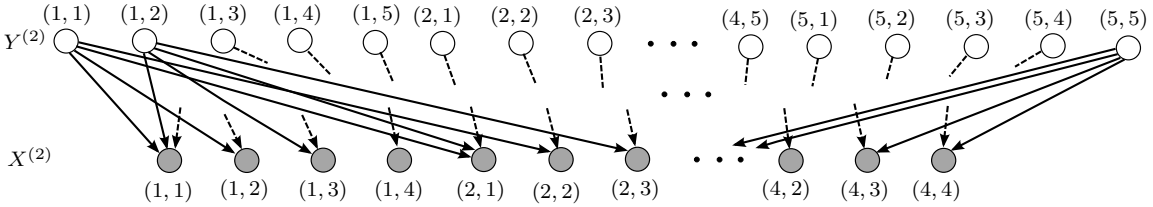
A real-valued order-n tensor $A \in \bigotimes_{i=1}^n \mathbb{R}^{p_i} := \mathbb{R}^{p_1 \times \cdots \times p_n}$ is a n dimensional array $A(1 : p_1, \ldots, 1 : p_n)$, where the i-th mode is indexed from 1 to p_i. In this paper, we restrict



(a) Structure of an overcomplete matrix $A \in \mathbb{R}^{4\times 5}$ having a perfect 2-gram matching.



(b) Structure of $A \odot A \in \mathbb{R}^{16\times 5}$ having a perfect (Y-saturating) matching, highlighted by dashed red edges.



(c) Structure of $A \otimes A \in \mathbb{R}^{16\times 25}$. For simplicity, only a few edges and nodes are shown and the dashed edges denote the bunch of edges connected to each node, not specifically shown.

Figure 4: An example of an overcomplete matrix A and the matrices $A \odot A$ and $A \otimes A$. The corresponding bipartite graphs encode the sparsity pattern of each of the matrices. $A \odot A$ expands the effect of hidden variables to second order observed variables which is crucial for overcomplete identifiability, while in the $A \otimes A$, the order of both the hidden and observed variables are increased.

ourselves to the case that $p_1 = \cdots = p_n = p$, and simply write $A \in \bigotimes^n \mathbb{R}^p$. A *fiber* of a tensor A is a vector obtained by fixing all indices of A except one, e.g., for $A \in \bigotimes^4 \mathbb{R}^3$, the vector $f = A(2, 1 : 3, 3, 1)$ is a fiber.
For a vector $u \in \mathbb{R}^p$, $\mathrm{Diag}_n(u) \in \bigotimes^n \mathbb{R}^p$ is the n-th order diagonal tensor with vector u on its diagonal. The tensor $A \in \bigotimes^n \mathbb{R}^p$, is stacked as a vector $a \in \mathbb{R}^{p^n}$ by the $\mathrm{vec}(\cdot)$ operator, defined as

$$a = \mathrm{vec}(A) \Leftrightarrow a\big((i_1 - 1)p^{n-1} + (i_2 - 1)p^{n-2} + \cdots + (i_{n-1} - 1)p + i_n\big) = A(i_1, i_2, \ldots, i_n).$$

The inverse of $a = \mathrm{vec}(A)$ operation is denoted by $A = \mathrm{ten}(a)$.
For vectors $a_i \in \mathbb{R}^{p_i}, i \in [n]$, the tensor outer product operator "$\circ$" is defined as (Golub and Loan, 2012)

$$A = a_1 \circ a_2 \circ \cdots \circ a_n \in \bigotimes_{i=1}^{n} \mathbb{R}^{p_i} \Leftrightarrow A(i_1, i_2, \ldots, i_n) := a_1(i_1)a_2(i_2) \cdots a_n(i_n). \tag{15}$$

The above generated tensor is a rank-1 tensor. The *tensor rank* is the minimal number of rank-1 tensors into which a tensor can be decomposed. This type of rank is called CP (Candecomp/Parafac) tensor rank in the literature (Golub and Loan, 2012).
According to above definitions, for any set of vectors $a_i \in \mathbb{R}^{p_i}, i \in [n]$, we have the following pair of equalities:

$$\mathrm{vec}(a_1 \circ a_2 \circ \cdots \circ a_n) = a_1 \otimes a_2 \otimes \cdots \otimes a_n,$$
$$\mathrm{ten}(a_1 \otimes a_2 \otimes \cdots \otimes a_n) = a_1 \circ a_2 \circ \cdots \circ a_n.$$

For any vector $a \in \mathbb{R}^p$, the power notations are also defined as

$$a^{\otimes n} := \overbrace{a \otimes a \otimes \cdots \otimes a}^{n \text{ times}} \in \mathbb{R}^{p^n},$$
$$a^{\circ n} := \overbrace{a \circ a \circ \cdots \circ a}^{n \text{ times}} \in \bigotimes^{n} \mathbb{R}^p.$$

The second power is usually called the n-th order *tensor power* of vector a.
Finally, the Tucker and CP (Candecomp/Parafac) representations are defined as follows (Golub and Loan, 2012; Kolda and Bader, 2009).

Definition 20 (Tucker representation) *Given a core tensor $S \in \bigotimes_{i=1}^n \mathbb{R}^{r_i}$ and inverse factors $U_i \in \mathbb{R}^{p_i \times r_i}, i \in [n]$, the Tucker representation of the n-th order tensor $A \in \bigotimes_{i=1}^n \mathbb{R}^{p_i}$ is*

$$A = \sum_{i_1=1}^{r_1} \sum_{i_2=1}^{r_2} \cdots \sum_{i_n=1}^{r_n} S(i_1, i_2, \ldots, i_n) U_1(:, i_1) \circ U_2(:, i_2) \circ \cdots \circ U_n(:, i_n) =: [[S; U_1, U_2, \ldots, U_n]], \tag{16}$$

where $U_j(:, i_j)$ denotes the i_j-th column of matrix U_j. The tensor S is referred to as the core tensor.

Definition 21 (CP representation) *Given $\lambda \in \mathbb{R}^r, U_i \in \mathbb{R}^{p_i \times r}, i \in [n]$, the CP representation of the n-th order tensor $A \in \bigotimes_{i=1}^n \mathbb{R}^{p_i}$ is*

$$A = \sum_{i=1}^{r} \lambda_i U_1(:,i) \circ U_2(:,i) \circ \cdots \circ U_n(:,i) =: [[\mathrm{Diag}_n(\lambda); U_1, U_2, \ldots, U_n]], \tag{17}$$

where $U_j(:,i)$ denotes the i-th column of matrix U_j.

Note that the CP representation is a special case of the Tucker representation when the core tensor S is square and diagonal.

4.2.2 Tensor Representation of Moments Under Topic Model

We now provide a tensor representation of the moments.

For the n-persistent topic model, the $2m$-th observed moment is denoted by $T_{2m}^{(n)}(x)$, which is the tensor form of the moment matrix $M_{2m}^{(n)}(x)$, characterized in Lemma 2. It is given by

$$T_{2m}(x)_{(i_1,i_2,\ldots,i_{2m})} := \mathbb{E}[x_1(i_1)x_2(i_2)\cdots x_{2m}(i_{2m})], \quad i_1, i_2, \ldots, i_{2m} \in [p], \tag{18}$$

where $T_{2m}(x) \in \bigotimes^{2m} \mathbb{R}^p$.

This tensor is characterized in the following lemma, and is proved in Appendix A.2.

Lemma 3 (n-persistent topic model moment characterization in tensor form) *The $(2m)$-th order moment of words, defined in equation (18), for the n-persistent topic model is characterized as*[12]*:*

- *if $m = rn$ for some integer $r \geq 1$, then*

$$\begin{aligned} T_{2m}^{(n)}(x) &= \sum_{i_1=1}^{q} \sum_{i_2=1}^{q} \cdots \sum_{i_{2r}=1}^{q} \mathbb{E}[h_{i_1} h_{i_2} \cdots h_{i_{2r}}] a_{i_1}^{\circ n} \circ a_{i_2}^{\circ n} \circ \cdots \circ a_{i_{2r}}^{\circ n} \\ &= \Big[\Big[S_r; \overbrace{A, A, \ldots, A}^{2m \text{ times}}\Big]\Big], \end{aligned} \tag{19}$$

  where $S_r \in \bigotimes^{2rn} \mathbb{R}^q$ is the core tensor in the above Tucker representation with the sparsity pattern as

$$S_r(\mathbf{i}) = \begin{cases} M_{2r}(h)_{\left((i_n, i_{2n}, \ldots, i_{rn}), (i_{(r+1)n}, i_{(r+2)n}, \ldots, i_{2rn})\right)} & , i_1 = i_2 = \cdots = i_n, i_{n+1} = i_{n+2} = \cdots = i_{2n}, \ldots \\ 0 & , \text{o. w.}, \end{cases}$$

  where $\mathbf{i} := (i_1, i_2, \ldots, i_{2rn})$.

- *If $n \geq 2m$, then*

$$T_{2m}^{(n)}(x) = \sum_{i \in [q]} \mathbb{E}[h_i] a_i^{\circ 2m} = \Big[\Big[\mathrm{Diag}_{2m}(\mathbb{E}[h]); \overbrace{A, A, \ldots, A}^{2m \text{ times}}\Big]\Big]. \tag{20}$$

12. The other cases not covered in Lemma 3 are deferred to Appendix A.2. See Remark 30.

The tensor representation in (19) is a specific type of tensor decomposition which is a special case of the Tucker representation (since S_r is not fully dense), but more general than the CP representation. The tensor representation in (20) has a CP form.

4.2.3 Comparison with Single Topic Model and Bag-of-words Admixture Model

We now provide the tensor form for the special cases single topic model and bag-of-words admixture model. In order to have a fair comparison, the number of observed variables is fixed to $2m$ and the persistence level is varied.

CP representation of the single topic model: The $(2m)$-th order moment of the words for the single topic model (infinite-persistent topic model) is provided in equation (20) as

$$T_{2m}^{(\infty)}(x) = \sum_{i\in[q]} \mathbb{E}[h_i] a_i^{\circ 2m} = \big[\big[\mathrm{Diag}_{2m}(\mathbb{E}[h]); \overbrace{A, A, \ldots, A}^{2m \text{ times}}\big]\big]. \tag{21}$$

This representation is the symmetric CP representation of $T_{2m}^{(\infty)}(x)$. In Appendix C, we provide a more detailed comparison between our approach and some of the previous identifiability results for the (overcomplete) CP decomposition. In particular, we show that our uniqueness result for CP decomposition is the sparse analogue of uniqueness result in Lathauwer (2006) where the factors of CP tensor decomposition (the columns of matrix A) satisfy specific sparsity constraints. See Appendix C for the details.

Tucker representation of the bag-of-words admixture model: From Lemma 3, the tensor form of the $(2m)$-th order moment of observed variables $x_l, l \in [2m]$, for the bag-of-words admixture model (1-persistent topic model) is given by

$$\begin{aligned} T_{2m}^{(1)}(x) &= \sum_{i_1=1}^{q} \sum_{i_2=1}^{q} \cdots \sum_{i_{2m}=1}^{q} \mathbb{E}[h_{i_1} h_{i_2} \cdots h_{i_{2m}}] a_{i_1} \circ a_{i_2} \circ \cdots \circ a_{i_{2m}} \\ &= \Big[\Big[\mathbb{E}\big[h^{\circ(2m)}\big]; \overbrace{A, A, \ldots, A}^{2m \text{ times}}\Big]\Big]. \end{aligned} \tag{22}$$

This representation is the Tucker representation (decomposition) of $T_{2m}^{(1)}(x)$ where the core tensor $S = \mathbb{E}\big[h^{\circ(2m)}\big]$ is the tensor form of the $(2m)$-th order hidden moment $M_{2m}(h)$, defined in equation (3), and the inverse factors correspond to the population structure A.

Comparing the tensor forms for the n-persistent topic model (19), single topic model (21), and bag of words admixture model (22), we find that all of them involve Tucker decompositions, where the inverse factors correspond to the topic-word matrix A, and the only difference is in the sparsity level of the core tensor S. For the bag of words model, with $n = 1$, the core tensor is fully dense in general, while for the single topic model, with $n \to \infty$, the core tensor is diagonal which reduces to the CP decomposition. For a general topic model with persistence level n, the core tensor is in between these two extremes and has structured sparsity. This sparsity property of the core tensor is crucial towards establishing identifiability in the overcomplete regime. The bag-of-words model is not identifiable in the overcomplete regime since the core tensor is fully dense in this case, while an overcomplete n-persistent topic model can be identified under certain constraints provided in Section 3, since the core tensor has structured sparsity and symmetry.

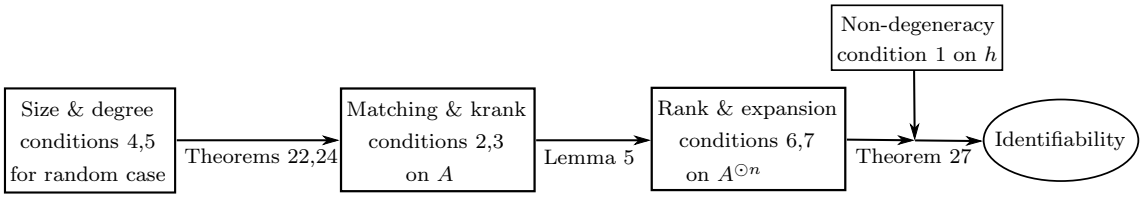


Figure 5: Hierarchy among the proposed conditions and results.

5. Proof Techniques and Auxiliary Results

The main identifiability results are given in Theorems 9 and 15 for deterministic and random cases of topic-word graph structures. In this section, we provide a proof sketch of these results, and then, we propose auxiliary results on the existence of perfect n-gram matching for random bipartite graphs and a lower bound on the Kruskal rank of random matrices.

5.1 Proof Sketch

Summary of relationships among different conditions: To summarize, there exists a hierarchy among the proposed conditions as follows. See Figure 5. First, in the random analysis, the size and the degree conditions 4 and 5 are sufficient for satisfying the perfect n-gram matching and the krank conditions 2 and 3, shown by Theorems 22 and 24. Then, these conditions 2 and 3 ensure that the rank and the expansion conditions 6 and 7 hold, shown by Lemma 5. And finally, these conditions 6 and 7 together with non-degeneracy condition 1 conclude the primary identifiability result in Theorem 27. Note that the genericity of A is also required for these results to hold.

Primary deterministic analysis in Theorem 27: The deterministic analysis is primarily based on conditions on the n-gram matrix $A^{\odot n}$; but since these conditions are opaque (mainly expansion condition on $A^{\odot n}$, provided in condition 7), this analysis is related to conditions on the matrix A itself (see Lemma 5). See Theorem 27 in Appendix A.1 for the identifiability result based on $A^{\odot n}$. We briefly discuss it below for the case when $2n$ words are available under the n-persistent topic model. From equation (8), the $(2n)$-th order moment of the observed variables under the n-persistent topic model can be written as

$$M_{2n}^{(n)}(x) = \Big(A^{\odot n}\Big)\mathbb{E}\big[hh^\top\big]\Big(A^{\odot n}\Big)^\top. \tag{23}$$

The question is whether we can recover A, given the $M_{2n}^{(n)}(x)$. Obviously, the matrix A is not identifiable without any further conditions. First, non-degeneracy and rank conditions (conditions 1 and 6) are required. Assuming these two conditions, we have from (23) that

$$\mathrm{Col}\Big(M_{2n}^{(n)}(x)\Big) = \mathrm{Col}\Big(A^{\odot n}\Big).$$

Therefore, the problem of recovering A from $M_{2n}^{(n)}(x)$ reduces to finding $A^{\odot n}$ in $\mathrm{Col}\big(A^{\odot n}\big)$. Then, we show that under the following expansion condition on $A^{\odot n}$ and the genericity

property, matrix A is identifiable from $\mathrm{Col}\big(A^{\odot n}\big)$. The expansion condition (refer to condition 7 for a more detailed statement), imposes the following property on the bipartite graph [13] $G\big(V_h, V_o^{(n)}; A^{\odot n}\big)$,

$$\Big|N_{A^{\odot n}_{\text{Rest.}}}(S)\Big| \geq |S| + d_{\max}\Big(A^{\odot n}\Big), \quad \forall S \subseteq V_h,\ |S| > \mathrm{krank}(A), \tag{24}$$

where $d_{\max}\big(A^{\odot n}\big)$ is the maximum node degree in set V_h, and the restricted version of n-gram matrix, denoted by $A^{\odot n}_{\text{Rest.}}$, is obtained by removing its redundant (identical) rows (see Definition 25). The identifiability claim is proved by showing that the columns of $A^{\odot n}$ are the sparsest and rank-1 vectors (in the tensor form) in $\mathrm{Col}\big(A^{\odot n}\big)$ under the expansion condition in (24) and genericity conditions. Note that since we only require expansion on sets larger than Kruskal rank, the expansion condition (24) is a more relaxed condition compared to expansion condition proposed in Anandkumar et al. (2012); Spielman et al. (2012a) for identifiability in the undercomplete regime. For a more detailed comparison, refer to Remark 26 in Appendix A.1.

Deterministic analysis in Theorem 9: Expansion and rank conditions in Theorem 27 are imposed on the n-gram matrix $A^{\odot n}$. According to the generalized matching notions, defined in Section 3.1, sufficient combinatorial conditions on matrix A (conditions 2 and 3) are introduced which ensure that the expansion and rank conditions on $A^{\odot n}$ are satisfied.

Recall Lemma 1 which says that existence of perfect n-gram matching in $G(Y, X; A)$ (condition 2) implies that $G(Y, X^{(n)}; A^{\odot n})$ has a perfect matching. Then, it is straightforward to argue that the expansion and rank conditions on $A^{\odot n}$ are satisfied, which is shown in Lemma 5 in Appendix A.3. This leads to the generic identifiability result stated in Theorem 9.

5.2 Analysis of Random Structures

The identifiability result for a random structured matrix A is provided in Theorem 15. Sufficient size and degree conditions 4 and 5 on the random matrix A are proposed such that the deterministic combinatorial conditions 2 and 3 on A are satisfied. The details of these auxiliary results are provided in the following two subsequent sections.[14] In Section 5.2.1, it is shown in Theorem 22 that a random bipartite graph satisfying reasonable size and degree constraints, has a perfect n-gram matching (condition 2), **whp**. Then, a lower bound on the Kruskal rank of a random matrix A under size and degree constraints is provided in Theorem 24 in Section 5.2.2, which implies the krank condition 3. Intuitions on why such size and degree conditions are required, are mentioned in Section 3.2 where these conditions are proposed.

5.2.1 Existence of Perfect n-gram Matching for Random Bipartite Graphs

We show in the following theorem that a random bipartite graph satisfying reasonable size and degree constraints, proposed earlier in conditions 4 and 5, has a perfect n-gram matching **whp**.

13. $V_o^{(n)}$ denotes all ordered n-tuples generated from set $V_o := \{1, \ldots, p\}$ which indexes the rows of $A^{\odot n}$.
14. Since these auxiliary results can also have independent interests as combinatorial results, we put them as theorems in the main part of the paper.

Theorem 22 (Existence of perfect n-gram matching for random bipartite graphs) *Consider a random bipartite graph $G(Y, X; E)$ with $|Y| = q$ nodes on the left side and $|X| = p$ nodes on the right side, and each node $i \in Y$ is randomly connected to d_i different nodes in X. Let $d_{\min} := \min_{i \in Y} d_i$. Assume that it satisfies the size condition $q \leq \left(c\frac{p}{n}\right)^n$ (condition 4) for some constant $0 < c < 1$ and the degree condition $d_{\min} \geq \max\{1 + \beta \log p, \alpha \log p\}$ for some constants $\beta > \frac{n-1}{\log 1/c}, \alpha > \max\left\{2n^2\left(\beta \log \frac{1}{c} + 1\right), 2\beta n\right\}$ (lower bound in condition 5). Then, there exists a perfect (Y-saturating) n-gram matching in the random bipartite graph $G(Y, X; E)$, with probability at least $1 - \gamma_1 p^{-\beta'}$ for constants $\beta' > 0$ and $\gamma_1 > 0$, specified in (5) and (6).*

See Appendix B.1 for the proof.

Note that the sufficient size bound $q = O(p^n)$ in the above theorem is also necessary (see Remark 7), and is therefore tight.

Remark 23 (Insufficiency of the union bound argument) *It is easier to exploit the union bound arguments to propose random bipartite graphs which have a perfect n-gram matching **whp**. It is proved in Appendix B.1 that if $d \geq n$ and the size constraint $|Y| = O(|X|^{\frac{n}{2}-\delta})$ for some $\delta > 0$ is satisfied, then **whp**, the random bipartite graph has a perfect n-gram matching. Comparing this result with ours in Theorem 22, our approach has a better size scaling while the union bound approach has a better degree scaling. The size scaling limitation in the union bound argument makes it unattractive. In order to identify the population structure A in the overcomplete regime where $|Y| = O(|X|^n)$, we need access to at least $(4n)$-th order moment under the union bound argument, while only the $(2n)$-th order moment is required under our argument.*

5.2.2 Lower Bound on the Kruskal Rank of Random Matrices

In the following theorem, a lower bound on the Kruskal rank of a random matrix A under dimension and degree constraints is provided.

Theorem 24 (Lower bound on the Kruskal rank of random matrices) *Consider a random matrix $A \in \mathbb{R}^{p \times q}$, where for any $i \in [q]$, there are d_i number of random non-zero entries in column i. Let $d_{\min} := \min_{i \in [q]} d_i$. Assume that it satisfies the size condition $q \leq \left(c\frac{p}{n}\right)^n$ (condition 4) for some constant $0 < c < 1$ and the degree condition $d_{\min} \geq 1 + \beta \log p$ for some constant $\beta > \frac{n-1}{\log 1/c}$ (lower bound in condition 5) and in addition A is generic. Then, $\operatorname{krank}(A) \geq cp$, with probability at least $1 - \gamma_2 p^{-\beta'}$ for constants $\beta' > 0$ and $\gamma_2 > 0$, specified in (5) and (7).*

See Appendix B.1 for the proof.

Acknowledgements

The authors acknowledge useful discussions with Sina Jafarpour, Adel Javanmard, Alex Dimakis, Moses Charikar, Sanjeev Arora, Ankur Moitra and Kamalika Chaudhuri. Sham Kakade thanks the Washington Research Foundation. A. Anandkumar is supported in part by Microsoft Faculty Fellowship, NSF Career award CCF-1254106, NSF Award CCF-1219234, ARO Award W911NF-12-1-0404, and ARO YIP Award W911NF-13-1-0084. M.

Janzamin is supported by NSF Award CCF-1219234, ARO Award W911NF-12-1-0404 and ARO YIP Award W911NF-13-1-0084.

Appendix A. Proof of Deterministic Identifiability Result (Theorem 9)

First, we show the identifiability result under an alternative set of conditions on the n-gram matrix, $A^{\odot n}$, and then, we show that the conditions of Theorem 9 are sufficient for these conditions to hold.

A.1 Deterministic Analysis Based on $A^{\odot n}$

In this section, the deterministic identifiability result based on conditions on the n-gram matrix, $A^{\odot n}$, is provided.

In the n-gram matrix, $A^{\odot n} \in \mathbb{R}^{p^n \times q}$, redundant rows exist. If some row of $A^{\odot n}$ is indexed by n-tuple $(i_1, \ldots, i_n) \in [p]^n$, then another row indexed by any permutation of the tuple $(i_1, \ldots, i_n)$ has the same entries. Therefore, the number of distinct rows of $A^{\odot n}$ is at most $\binom{p+n-1}{n}$. In the following definition, we define a non-redundant version of n-gram matrix which is restricted to the (potentially) distinct rows.

Definition 25 (Restricted n-gram matrix) *For any matrix $A \in \mathbb{R}^{p \times q}$, restricted n-gram matrix $A^{\odot n}_{\text{Rest.}} \in \mathbb{R}^{s \times q}$, $s = \binom{p+n-1}{n}$, is defined as the restricted version of n-gram matrix $A^{\odot n} \in \mathbb{R}^{p^n \times q}$, where the redundant rows of $A^{\odot n}$ are removed, as explained above.*

Condition 6 (Rank condition) *The n-gram matrix $A^{\odot n}$ is full column rank.*

Condition 7 (Graph expansion) *Let $G(V_h, V_o^{(n)}; A^{\odot n})$ denote the bipartite graph with vertex sets V_h corresponding to the hidden variables (indexing the columns of $A^{\odot n}$) and $V_o^{(n)}$ corresponding to the n-th order observed variables (indexing the rows of $A^{\odot n}$) and edge matrix $A^{\odot n} \in \mathbb{R}^{|V_o^{(n)}| \times |V_h|}$. The bipartite graph $G(V_h, V_o^{(n)}; A^{\odot n})$ satisfies the following* expansion property [15] *on the restricted version specified by $A^{\odot n}_{\text{Rest.}}$,*

$$\left| N_{A^{\odot n}_{\text{Rest.}}}(S) \right| \geq |S| + d_{\max}\left(A^{\odot n}\right), \quad \forall S \subseteq V_h,\ |S| > \text{krank}(A), \tag{25}$$

where $d_{\max}\left(A^{\odot n}\right)$ is the maximum node degree in set V_h.

Remark 26 *The expansion condition for the bag-of-words admixture model is provided in* (4)*, introduced in Anandkumar et al. (2012). The proposed expansion condition in* (25) *is inherited from* (4)*, with two major modifications. First, the condition is appropriately generalized for our model which involves a graph with edges specified by the n-gram matrix,*

15. Note that this notion of generalized expansion is different from unbalanced expander graphs proposed in the compressed sensing literature (Khajehnejad et al., 2011; Indyk and Razenshteyn, 2013). For a left regular bipartite graph $G(Y, X; A)$ with regular degree d for the vertices on Y side, we say that it is a (k, ϵ)-expander if for any set $S \subseteq Y$ with $|S| \leq k$, we have $N_A(S) \geq |S|d(1-\epsilon)$. This is completely different with the expansion condition we define here in some aspects: first our expansion condition is additive while this one is multiplicative, and second our expansion condition is imposed on large sets while this one is imposed on small sets.

$A^{\odot n}$, *as stated in* (23). *Second, the expansion property* (4), *proposed in Anandkumar et al. (2012), needs to be satisfied for all subsets* S *with size* $|S| \geq 2$, *which is a stricter condition than the one proposed here in* (25), *since we can have* $\text{krank}(A) \gg 2$.

The deterministic identifiability result based on the conditions on $A^{\odot n}$, is stated in the following theorem for $n \geq 2$, while $n = 1$ case is addressed in Remarks 10 and 26. The identifiability result relies on access to the $(2n)$-th order moment of observed variables $x_l, l \in [2n]$, defined in equation (2) as

$$M_{2n}(x) := \mathbb{E}\left[(x_1 \otimes x_2 \otimes \cdots \otimes x_n)(x_{n+1} \otimes x_{n+2} \otimes \cdots \otimes x_{2n})^\top\right] \in \mathbb{R}^{p^n \times p^n}.$$

Theorem 27 (Generic identifiability under deterministic conditions on $A^{\odot n}$) *Let* $M_{2n}^{(n)}(x)$ *(defined in equation* (2)*) be the* $(2n)$*-th order moment of the* n*-persistent topic model described in Section 2. If the model satisfies conditions 1, 6 and 7, then, for any* $n \geq 2$, *all the columns of population structure* A *are* generically identifiable *from* $M_{2n}^{(n)}(x)$.

Proof: Define $B := A^{\odot n} \in \mathbb{R}^{p^n \times q}$. Then, the moment characterized in equation (23) can be written as $M_{2n}^{(n)}(x) = B\mathbb{E}\left[hh^\top\right] B^\top$. Since both matrices $\mathbb{E}\left[hh^\top\right]$ and B have full column rank (from conditions 1 and 6), the rank of $B\mathbb{E}\left[hh^\top\right] B^\top$ is q where $q = O(p^n)$, and furthermore $\text{Col}(B\mathbb{E}\left[hh^\top\right] B^\top) = \text{Col}(B)$. Let $\mathcal{U} := \{u_1, \ldots, u_q\} \in \mathbb{R}^{p^n}$ be any basis of $\text{Col}(B\mathbb{E}\left[hh^\top\right] B^\top)$ satisfying the following two properties:

1) The maximum of ℓ_0 norm of u_i's is minimized (among all basis sets).

2) The tensor rank of u_i's (in the n-th order tensor form) is equal to 1, i.e., $\text{Rank}(\text{ten}(u_i)) = 1, i \in [q]$.

Let the columns of matrix B be b_i for $i \in [q]$. Since all the b_i's (which belong to $\text{Col}(B\mathbb{E}\left[hh^\top\right] B^\top)$) are rank-1 in the n-th order tensor form (since $\text{ten}(b_i) = a_i^{\circ n}$) and the number of non-zero entries in each of b_i's is at most $d_{\max}(B) = d_{\max}(A)^n$, we conclude that

$$\max_i \text{Rank}(\text{ten}(u_i)) = 1 \quad \text{and} \quad \max_i \|u_i\|_0 \leq d_{\max}(B). \tag{26}$$

The above bounds are concluded from the fact that $b_i \in \text{Col}(B\mathbb{E}\left[hh^\top\right] B^\top)$, $i \in [q]$, and therefore the ℓ_0 norm and the rank properties of b_i's are upper bounds for the corresponding properties of basis vectors u_i's (according to the proposed conditions for u_i's).
Now, exploiting these observations and also the genericity of A and the expansion condition 7, we show that the basis vectors u_i's are scaled columns of B. Since u_i for $i \in [q]$, is a vector in the column space of B, it can be represented as $u_i = Bv_i$ for some vector $v_i \in \mathbb{R}^q$. Equivalently, for any $i \in [q]$, $u_i = \sum_{j=1}^q v_i(j) b_j$ where $b_j = a_j^{\otimes n}$ is the j-th column of matrix B and $v_i(j)$ is a scalar which is the j-th entry of vector v_i. Then, the tensor form of u_i can be written as

$$\text{ten}(u_i) = \sum_{j=1}^q v_i(j)\, \text{ten}(b_j) = \sum_{j=1}^q v_i(j)\, \text{ten}(a_j^{\otimes n}) = \sum_{j=1}^q v_i(j) a_j^{\circ n} = [[\text{Diag}_n(v_i); \overbrace{A, \ldots, A}^{n \text{ times}}]], \tag{27}$$

where the last equality is based on the notation defined in Definition 21, and $\mathrm{Diag}_n(v_i)$ is defined as the n-th order diagonal tensor with vector v_i on its diagonal. We define $\widetilde{v}_i := [v_i(j)]_{j:v_i(j)\neq 0}$ as the vector which contains only the non-zero entries of v_i, i.e., $\widetilde{v}_i$ is the restriction of vector v_i to its support. Therefore, $\widetilde{v}_i \in \mathbb{R}^r$, where $r := \|v_i\|_0$. Furthermore, the matrix $\widetilde{A}_i := \{a_j : v_i(j) \neq 0\} \in \mathbb{R}^{p\times r}$ is defined as the restriction of A to its columns corresponding to the support of v_i. Let $(\widetilde{a}_i)_j$ denote the j-th column of $\widetilde{A}_i$. According to these definitions, equation (27) reduces to

$$\mathrm{ten}(u_i) = [[\mathrm{Diag}_n(\widetilde{v}_i); \overbrace{\widetilde{A}_i, \ldots, \widetilde{A}_i}^{n \text{ times}}]] = \sum_{j=1}^{r} \widetilde{v}_i(j)[(\widetilde{a}_i)_j]^{\circ n}, \tag{28}$$

which is derived by removing columns of A corresponding to the zero entries in v_i.
Next, we rule out that $\|v_i\|_0 \geq 2$ under two cases ($2 \leq \|v_i\|_0 \leq \mathrm{krank}(A)$ and $\mathrm{krank}(A) < \|v_i\|_0 \leq q$), to conclude that u_i's vectors are scaled columns of B.

Case 1: $2 \leq \|v_i\|_0 \leq \mathrm{krank}(A)$**.** Here, the number of columns of $\widetilde{A}_i \in \mathbb{R}^{p\times\|v_i\|_0}$ is less than or equal to $\mathrm{krank}(A)$ and therefore it is full column rank. Since, all the components of CP representation in equation (28) are full column rank [16], for any [17] $n \geq 2$, we have $\mathrm{Rank}(\mathrm{ten}(u_i)) = r = \|v_i\|_0 > 1$, which contradicts the fact that $\max_i \mathrm{Rank}(\mathrm{ten}(u_i)) = 1$ in (26).

Note that for the *full column rank* topic-word matrix $A \in \mathbb{R}^{p\times q}$ (where $\mathrm{Rank}(A) = \mathrm{krank}(A) = q$) as in Corollary 13, it is sufficient to argue this case and there is no need to argue next case. This is why the expansion condition is not required in Corollary 13.

Case 2: $\mathrm{krank}(A) < \|v_i\|_0 \leq q$**.** Here, we first restrict the n-gram matrix B to distinct rows, denoted by $B_{\mathrm{Rest.}}$, as defined in Definition 25. Let $u_i' = B_{\mathrm{Rest.}}v_i$. Since u_i' is the restricted version of u_i, we have

$$\begin{aligned}\|u_i\|_0 \geq \|u_i'\|_0 &= \|B_{\mathrm{Rest.}}v_i\|_0 \\ &> \left|N_{B_{\mathrm{Rest.}}}(\mathrm{Supp}(v_i))\right| - |\,\mathrm{Supp}(v_i)| \\ &\geq d_{\max}(B),\end{aligned}$$

where the second inequality is from Lemma 4 (which is stated and proved right after this theorem), and the third inequality follows from the graph expansion property (condition 7). This result contradicts the fact that $\max_i \|u_i\|_0 \leq d_{\max}(B)$ in (26).

From above contradictions, $\|v_i\|_0 = 1$ and hence, columns of $B := A^{\odot n}$ are the scaled versions of u_i's. ■

The following lemma is useful in the proof of Theorem 27. The result proposed in this lemma is similar to the parameter genericity condition in Anandkumar et al. (2012), but generalized for the n-gram matrix, $A^{\odot n}$. The lemma is proved along the lines of the proof of Remark 2.2 in Anandkumar et al. (2012).

16. Note that for $n \geq 3$, this full rank condition can be relaxed by Kruskal's condition for uniqueness of CP decomposition (Kruskal, 1977) and its generalization to higher order tensors (Sidiropoulos and Bro, 2000). Precisely, instead of saying $\mathrm{Rank}(\widetilde{A}_i) = \mathrm{krank}(\widetilde{A}_i) = r$, it is only required to have $\mathrm{krank}(\widetilde{A}_i) \geq (2r+n-1)/n$ to argue the result of case 1. This only improves the constants involved in the final result.
17. Note that for $n = 1$, since the (tensor) rank of any vector is 1, this analysis does not work.

Lemma 4 *If $A \in \mathbb{R}^{p\times q}$ is generic (see Definition 2), then the n-gram matrix $A^{\odot n} \in \mathbb{R}^{p^n\times q}$ satisfies the following property with Lebesgue measure one. For any vector $v \in \mathbb{R}^q$ with $\|v\|_0 \geq 2$, we have*

$$\left\|A^{\odot n}_{\text{Rest.}} v\right\|_0 > \left|N_{A^{\odot n}_{\text{Rest.}}}(\text{Supp}(v))\right| - |\,\text{Supp}(v)|,$$

where for a set $S \subseteq [q]$, $N_{A^{\odot n}}(S) := \{i \in [p]^n : A^{\odot n}(i,j) \neq 0 \text{ for some } j \in S\}$.

Here, we prove the result for the case of $n = 2$. The proof can be easily generalized to larger n.

Let $A := P + Z$ be generic, where P is an arbitrary matrix perturbed by random continuous independent [18] perturbations Z. Consider the 2-gram matrix $B := A \odot A \in \mathbb{R}^{p^2\times q}$. We show that the restricted version of B, denoted by $\widetilde{B} := B_{\text{Rest.}} \in \mathbb{R}^{\frac{p(p+1)}{2}\times q}$, satisfies the above genericity condition. Before that, we first establish some definitions and one claim.

Definition 28 *We call a vector* fully dense *if all of its entries are non-zero.*

Definition 29 *We say a matrix has the* Null Space Property (NSP) *if its null space does not contain any fully dense vector.*

Claim 1 *Fix any $S \subseteq [q]$ with $|S| \geq 2$, and set $R := N_{(P^{\odot 2})_{\text{Rest.}}}(S)$. Let $\widetilde{C}$ be a $|S| \times |S|$ submatrix of $\widetilde{B}_{R,S}$. Then* $\Pr(\widetilde{C} \text{ has the NSP}) = 1$.

Proof of Claim 1: First, note that $\widetilde{B}$ can be expanded as

$$\widetilde{B} := (A \odot A)_{\text{Rest.}} = (P \odot P)_{\text{Rest.}} + \underbrace{(P \odot Z + Z \odot P)_{\text{Rest.}} + (Z \odot Z)_{\text{Rest.}}}_{:=U}.$$

Let $s = |S|$ and let $\widetilde{C} = [\tilde{c}_1|\tilde{c}_2|\cdots|\tilde{c}_s]^\top$, where $\tilde{c}_i^\top$ is the i-th row of $\widetilde{C}$. Also, let $C := [c_1|c_2|\cdots|c_s]^\top$ and $W := [w_1|w_2|\cdots|w_s]^\top$ be the corresponding $|S| \times |S|$ submatrices of $\left(P^{\odot 2}\right)_{\text{Rest.}}$ and U, respectively. For each $i \in [s]$, denote by $\mathcal{N}_i$ the null space of the matrix $\widetilde{C}_i = [\tilde{c}_1|\tilde{c}_2|\cdots|\tilde{c}_i]^\top$. Finally let $\mathcal{N}_0 = \mathbb{R}^s$. Then, $\mathcal{N}_0 \supseteq \mathcal{N}_1 \supseteq \cdots \supseteq \mathcal{N}_s$. We need to show that, with probability one, $\mathcal{N}_s$ does not contain any fully dense vector.

If one of $\mathcal{N}_i, i \in [s]$, does not contain any full dense vector, the result is proved. Suppose that $\mathcal{N}_i$ contains some fully dense vector v. Since C is a submatrix of $\left(P^{\odot 2}\right)_{R,S}$, every row $c_{i+1}^\top$ of C contains at least one non-zero entry. Therefore,

$$\begin{aligned} v^\top \tilde{c}_{i+1} &= \sum_{j\in[s]} v(j)\tilde{c}_{i+1}(j) \\ &= \sum_{j\in[s]:c_{i+1}(j)\neq 0} v(j)(c_{i+1}(j) + w_{i+1}(j)), \end{aligned}$$

18. Note that the distribution of Z does not matter as long as the independence and continuous conditions hold.

where $\{w_{i+1}(j) : j \in [s] \text{ s.t. } c_{i+1}(j) \neq 0\}$ are independent random variables, and moreover, they are independent of $\tilde{c}_1, \ldots, \tilde{c}_i$ and thus of v. By assumption on the distribution of the $w_{i+1}(j)$,

$$\Pr\left[v \in \mathcal{N}_{i+1} \middle| \tilde{c}_1, \tilde{c}_2, \ldots, \tilde{c}_i\right] = \Pr\left[\sum_{j \in [s]: c_{i+1}(j) \neq 0} v(j)(c_{i+1}(j) + w_{i+1}(j)) = 0 \middle| \tilde{c}_1, \tilde{c}_2, \ldots, \tilde{c}_i\right] = 0. \quad (29)$$

Consequently,

$$\Pr\left[\dim(\mathcal{N}_{i+1}) < \dim(\mathcal{N}_i) \middle| \tilde{c}_1, \tilde{c}_2, \ldots, \tilde{c}_i\right] = 1 \quad (30)$$

for all $i = 0, \ldots, s-1$. As a result, with probability one, $\dim(\mathcal{N}_s) = 0$. ■

Now, we are ready to prove Lemma 4.

Proof of Lemma 4: It follows from Claim 1 that, with probability one, the following event holds: for every $S \subseteq [q], |S| \geq 2$, and every $|S| \times |S|$ submatrix $\widetilde{C}$ of $\widetilde{B}_{R,S}$ where $R := N_{(P^{\odot 2})_{\text{Rest.}}}(S)$, then $\widetilde{C}$ has the NSP.

Now fix $v \in \mathbb{R}^q$ with $\|v\|_0 \geq 2$. Let $S := \text{Supp}(v)$ and $H := \widetilde{B}_{R,S}$. Furthermore, let $u \in (\mathbb{R} \setminus \{0\})^{|S|}$ be the restriction of vector v to S; observe that u is fully dense. It is clear that $\|\widetilde{B}v\|_0 = \|Hu\|_0$, so we need to show that

$$\|Hu\|_0 > |R| - |S|. \quad (31)$$

For the sake of contradiction, suppose that Hu has at most $|R| - |S|$ non-zero entries. Since $Hu \in \mathbb{R}^{|R|}$, there is a subset of $|S|$ entries on which Hu is zero. This corresponds to a $|S| \times |S|$ submatrix of $H := \widetilde{B}_{R,S}$ which contains u in its null space. It means that this submatrix does not have the NSP, which is a contradiction. Therefore we conclude that Hu must have more than $|R| - |S|$ non-zero entries, which finishes the proof. ■

A.2 Proof of Moment Characterization Lemmata

Remark 30 *In Lemmata 2 and 3, a specific case of order and persistence ($m = rn$) was considered. Here, we provide the moment form for a more general case. Assume that $m = rn + s$ for some integers $r \geq 1, 1 \leq s \leq \frac{n}{2}$, then*

$$M_{2m}^{(n)}(x) = \left(\overbrace{A^{\odot n} \otimes \cdots \otimes A^{\odot n}}^{r \text{ times}} \otimes A^{\odot s}\right) \widetilde{M}_{2r}(h) \left(A^{\odot(n-s)} \otimes \overbrace{A^{\odot n} \otimes \cdots \otimes A^{\odot n}}^{r-1 \text{ times}} \otimes A^{\odot(2s)}\right)^\top,$$

where $\widetilde{M}_{2r}(h) \in \mathbb{R}^{q^{r+1} \times q^{r+1}}$ is the hidden moment as

$$\widetilde{M}_{2r}(h)_{\left((i_1,\ldots,i_{r+1}),(j_1,\ldots,j_{r+1})\right)} := \begin{cases} \mathbb{E}[h_{i_1} \cdots h_{i_r} h_{i_{r+1}}^2 h_{j_2} \cdots h_{j_{r+1}}] & \text{if } i_{r+1} = j_1, \\ 0 & \text{o. w .} \end{cases}$$

The tensor form is also characterized as

$$T_{2m}^{(n)}(x) = \Big[\Big[\widetilde{S}_r; \overbrace{A, A, \ldots, A}^{2m \text{ times}}\Big]\Big],$$

where $\widetilde{S}_r \in \bigotimes^{2m} \mathbb{R}^q$ *is the core tensor in the above Tucker representation with the sparsity pattern as follows. Let* $\mathbf{i} := (i_1, i_2, \ldots, i_{2m})$. *If*

$$i_1 = i_2 = \cdots = i_n, i_{n+1} = i_{n+2} = \cdots = i_{2n}, \cdots, i_{(2r-1)n+1} = i_{(2r-1)n+2} = \cdots = i_{2rn},$$
$$i_{2(m-s)+1} = i_{2(m-s)+2} = \cdots = i_{2m},$$

we have

$$\widetilde{S}_r(\mathbf{i}) = \widetilde{M}_{2r}(h)_{\left((i_n, i_{2n}, \ldots, i_{rn}, i_m), (i_{(r+1)n}, i_{(r+2)n}, \ldots, i_{2rn}, i_{2m})\right)}.$$

Otherwise, $\widetilde{S}_r(\mathbf{i}) = 0$.

Proof of Lemma 2: The proof is basically incorporating the conditional independence relationships between random variables x_l and y_j under the n-persistent topic model.

In order to simplify the notation, similar to tensor powers for vectors, the tensor power for a matrix $U \in \mathbb{R}^{p \times q}$ is defined as

$$U^{\otimes r} := \overbrace{U \otimes U \otimes \cdots \otimes U}^{r \text{ times}} \in \mathbb{R}^{p^r \times q^r}. \tag{32}$$

First, consider the case $m = rn$ for some integer $r \geq 1$. One advantage of encoding $y_j, j \in [2r]$, by basis vectors appears in characterizing the conditional moments. The first order conditional moment of words $x_l, l \in [2m]$, in the n-persistent topic model can be written as

$$\mathbb{E}\big[x_{(j-1)n+k} | y_j\big] = A y_j, \ j \in [2r], \ k \in [n],$$

where $A = [a_1 | a_2 | \cdots | a_q] \in \mathbb{R}^{p \times q}$. Next, the m-th order conditional moment of different views $x_l, l \in [m]$, in the n-persistent topic model can be written as

$$\mathbb{E}[x_1 \otimes x_2 \otimes \cdots \otimes x_m | y_1 = e_{i_1}, y_2 = e_{i_2}, \ldots, y_r = e_{i_r}] = a_{i_1}^{\otimes n} \otimes a_{i_2}^{\otimes n} \otimes \cdots \otimes a_{i_r}^{\otimes n},$$

which is derived from the conditional independence relationships among the observations $x_l, l \in [m]$, given topics $y_j, j \in [r]$. Similar to the first order moments, since vectors $y_j, j \in [r]$, are encoded by the basis vectors $e_i \in \mathbb{R}^q$, the above moment can be written as the following matrix multiplication

$$\mathbb{E}[x_1 \otimes x_2 \otimes \cdots \otimes x_m | y_1, y_2, \ldots, y_r] = \left(A^{\odot n}\right)^{\otimes r} (y_1 \otimes y_2 \otimes \cdots \otimes y_r), \tag{33}$$

where the $(\cdot)^{\otimes r}$ notation is defined in equation (32). Now for the $(2m)$-th order moment, we have

$$M_{2m}^{(n)}(x) := \mathbb{E}\Big[(x_1 \otimes x_2 \otimes \cdots \otimes x_m)(x_{m+1} \otimes x_{m+2} \otimes \cdots \otimes x_{2m})^\top\Big]$$

$$
\begin{aligned}
&= \mathbb{E}_{(y_1,y_2,\ldots,y_{2r})}\Big[\mathbb{E}\left[(x_1 \otimes \cdots \otimes x_m)(x_{m+1} \otimes \cdots \otimes x_{2m})^\top | y_1, y_2, \ldots, y_{2r}\right]\Big] \\
&\overset{(a)}{=} \mathbb{E}_{(y_1,y_2,\ldots,y_{2r})}\Big[\mathbb{E}\big[(x_1 \otimes \cdots \otimes x_m)|y_1,\ldots,y_{2r}\big]\mathbb{E}\big[(x_{m+1} \otimes \cdots \otimes x_{2m})^\top |y_1,\ldots,y_{2r}\big]\Big] \\
&\overset{(b)}{=} \mathbb{E}_{(y_1,y_2,\ldots,y_{2r})}\Big[\mathbb{E}\big[(x_1 \otimes \cdots \otimes x_m)|y_1,\ldots,y_r\big]\mathbb{E}\big[(x_{m+1} \otimes \cdots \otimes x_{2m})^\top |y_{r+1},\ldots,y_{2r}\big]\Big] \\
&\overset{(c)}{=} \mathbb{E}_{(y_1,y_2,\ldots,y_{2r})}\left[\left(\left[A^{\odot n}\right]^{\otimes r}\right)(y_1 \otimes \cdots \otimes y_r)(y_{r+1} \otimes \cdots \otimes y_{2r})^\top \left(\left[A^{\odot n}\right]^{\otimes r}\right)^\top\right] \\
&= \left(\left[A^{\odot n}\right]^{\otimes r}\right)\mathbb{E}\left[(y_1 \otimes \cdots \otimes y_r)(y_{r+1} \otimes \cdots \otimes y_{2r})^\top\right]\left(\left[A^{\odot n}\right]^{\otimes r}\right)^\top \\
&\overset{(d)}{=} \left(\left[A^{\odot n}\right]^{\otimes r}\right)M_{2r}(y)\left(\left[A^{\odot n}\right]^{\otimes r}\right)^\top, \qquad (34)
\end{aligned}
$$

where (a) results from the independence of $(x_1,\ldots,x_m)$ and $(x_{m+1},\ldots,x_{2m})$ given $(y_1,y_2,\ldots,y_{2r})$ and (b) is concluded from the independence of $(x_1,\ldots,x_m)$ and $(y_{r+1},\ldots,y_{2r})$ given $(y_1,\ldots,y_r)$ and the independence of $(x_{m+1},\ldots,x_{2m})$ and $(y_1,\ldots,y_r)$ given $(y_{r+1},\ldots,y_{2r})$. Equation (33) is used in (c) and finally, the $(2r)$-th order moment of $(y_1,\ldots,y_{2r})$ is defined as $M_{2r}(y) := \mathbb{E}\left[(y_1 \otimes \cdots \otimes y_r)(y_{r+1} \otimes \cdots \otimes y_{2r})^\top\right]$ in (d).

For $M_{2r}(y)$, we have by the law of total expectation

$$
\begin{aligned}
M_{2r}(y) &:= \mathbb{E}\big[(y_1 \otimes \cdots \otimes y_r)(y_{r+1} \otimes \cdots \otimes y_{2r})^\top\big] \\
&= \mathbb{E}_h\Big[\mathbb{E}\big[(y_1 \otimes \cdots \otimes y_r)(y_{r+1} \otimes \cdots \otimes y_{2r})^\top |h\big]\Big] \\
&= \mathbb{E}_h\left[\Big(\overbrace{h \otimes \cdots \otimes h}^{r \text{ times}}\Big)\Big(\overbrace{h \otimes \cdots \otimes h}^{r \text{ times}}\Big)^\top\right] \\
&= M_{2r}(h),
\end{aligned}
$$

where the third equality is concluded from the conditional independence of variables $y_j, j \in [2r]$, given h and the model assumption that $\mathbb{E}\big[y_j|h\big] = h, j \in [2r]$. Substituting this in equation (34), finishes the proof for the n-persistent topic model. Similarly, the moment of single topic model (infinite persistence) can be also derived. ■

Proof of Lemma 3: Defining $\Lambda := M_{2r}(h) \in \mathbb{R}^{q^r \times q^r}$ and $B := \big[A^{\odot n}\big]^{\otimes r} \in \mathbb{R}^{p^{rn} \times q^r}$, the $(2rn)$-th order moment $M_{2rn}^{(n)}(x) \in \mathbb{R}^{p^{rn} \times p^{rn}}$ of the n-persistent topic model proposed in equation (8) can be written as

$$M_{2rn}^{(n)}(x) = B\Lambda B^\top.$$

Let $b_{(i_1,\ldots,i_r)} \in \mathbb{R}^{p^{rn}}$ denote the corresponding column of B indexed by r-tuple $(i_1,\ldots,i_r), i_k \in [q], k \in [r]$. Then, the above matrix equation can be expanded as

$$
\begin{aligned}
M_{2rn}^{(n)}(x) &= \sum_{\substack{i_1,\ldots,i_r \in [q] \\ j_1,\ldots,j_r \in [q]}} \Lambda\big((i_1,\ldots,i_r),(j_1,\ldots,j_r)\big) b_{(i_1,\ldots,i_r)} b_{(j_1,\ldots,j_r)}^\top \\
&= \sum_{\substack{i_1,\ldots,i_r \in [q] \\ j_1,\ldots,j_r \in [q]}} \Lambda\big((i_1,\ldots,i_r),(j_1,\ldots,j_r)\big) [a_{i_1}^{\otimes n} \otimes \cdots \otimes a_{i_r}^{\otimes n}][a_{j_1}^{\otimes n} \otimes \cdots \otimes a_{j_r}^{\otimes n}]^\top,
\end{aligned}
$$

where relation $b_{(i_1,\ldots,i_r)} = a_{i_1}^{\otimes n} \otimes \cdots \otimes a_{i_r}^{\otimes n}, i_1, \ldots, i_r \in [q]$, is used in the last equality. Let $m_{2rn}^{(n)}(x) \in \mathbb{R}^{p^{2rn}}$ denote the vectorized form of $(2rn)$-th order moment $M_{2rn}^{(n)}(x) \in \mathbb{R}^{p^{rn} \times p^{rn}}$. Therefore, we have

$$\begin{aligned} m_{2rn}^{(n)}(x) &:= \operatorname{vec}\Big(M_{2rn}^{(n)}(x)\Big) \\ &= \sum_{\substack{i_1,\ldots,i_r \in [q] \\ j_1,\ldots,j_r \in [q]}} \Lambda\big((i_1,\ldots,i_r),(j_1,\ldots,j_r)\big) a_{i_1}^{\otimes n} \otimes \cdots \otimes a_{i_r}^{\otimes n} \otimes a_{j_1}^{\otimes n} \otimes \cdots \otimes a_{j_r}^{\otimes n}. \end{aligned}$$

Then, we have the following equivalent tensor form for the original model proposed in equation (8)

$$\begin{aligned} T_{2rn}^{(n)}(x) &:= \operatorname{ten}\Big(m_{2rn}^{(n)}(x)\Big) \\ &= \sum_{\substack{i_1,\ldots,i_r \in [q] \\ j_1,\ldots,j_r \in [q]}} \Lambda\big((i_1,\ldots,i_r),(j_1,\ldots,j_r)\big) a_{i_1}^{\circ n} \circ \cdots \circ a_{i_r}^{\circ n} \circ a_{j_1}^{\circ n} \circ \cdots \circ a_{j_r}^{\circ n}. \end{aligned}$$

■

A.3 Sufficient Matching Properties for Satisfying Rank and Graph Expansion Conditions

In the following lemma, it is shown that under a perfect n-gram matching and additional genericity and krank conditions, the rank and graph expansion conditions 6 and 7 on $A^{\odot n}$, are satisfied.

Lemma 5 *Assume that the bipartite graph $G(V_h, V_o; A)$ has a perfect n-gram matching (condition 2 is satisfied). Then, the following results hold for the n-gram matrix $A^{\odot n}$:*

1) *If A is generic, $A^{\odot n}$ is full column rank (condition 6) with Lebesgue measure one (almost surely).*

2) *If krank condition 3 holds, $A^{\odot n}$ satisfies the proposed expansion property in condition 7.*

Proof: Let M denote the perfect n-gram matching of the bipartite graph $G(V_h, V_o; A)$. From Lemma 1, there exists a perfect matching $M^{\odot n}$ for the bipartite graph $G(V_h, V_o^{(n)}; A^{\odot n})$. Denote the corresponding bi-adjacency matrix to the edge set M as A_M. Similarly, B_M denotes the corresponding bi-adjacency matrix to the edge set $M^{\odot n}$. Note that $\operatorname{Supp}(A_M) \subseteq \operatorname{Supp}(A)$ and $\operatorname{Supp}(B_M) \subseteq \operatorname{Supp}(A^{\odot n})$.

Since B_M is a perfect matching, it consists of $q := |V_h|$ rows, each of which has only one non-zero entry, and furthermore, the non-zero entries are in q different columns. Therefore, these rows form q linearly independent vectors. Since the row rank and column rank of a matrix are equal, and the number of columns of B_M is q, the column rank of B_M is q or in other words, B_M is full column rank. Since A is generic, from Lemma 6 (with a slight

modification in the analysis [19]), $A^{\odot n}$ is also full column rank with Lebesgue measure one (almost surely). This completes the proof of part 1.

Next, we prove the second part. From krank definition, we have

$$|N_A(S')| \geq |S'| \quad \text{for } S' \subseteq V_h, |S'| \leq \text{krank}(A),$$

which is concluded from the fact that the corresponding submatrix of A specified by S' should be full column rank. From this inequality, we have

$$|N_A(S')| \geq \text{krank}(A) \quad \text{for } S' \subseteq V_h, |S'| = \text{krank}(A). \tag{35}$$

Then, we have

$$\begin{aligned} |N_A(S)| &\geq |N_A(S')| \quad \text{for } S' \subset S \subseteq V_h, |S| > \text{krank}(A), |S'| = \text{krank}(A), \\ &\geq \text{krank}(A) \\ &\geq d_{\max}(A)^n, \end{aligned} \tag{36}$$

where (35) is used in the second inequality and the last inequality is from krank condition 3.

In the restricted n-gram matrix $A^{\odot n}_{\text{Rest.}}$, the number of neighbors for a set $S \subseteq V_h, |S| > \text{krank}(A)$, can be bounded as

$$\begin{aligned} \left|N_{A^{\odot n}_{\text{Rest.}}}(S)\right| &\geq |N_A(S)| + |S| \\ &\geq d_{\max}(A)^n + |S| \quad \text{for } |S| > \text{krank}(A), \end{aligned} \tag{37}$$

where the first inequality is due to the fact that the set $N_{A^{\odot n}_{\text{Rest.}}}$ consists of rows indexed by the following two [20] subsets: n-tuples $(i, i, \ldots, i)$ where all the indices are equal and n-tuples $(i_1, \ldots, i_n)$ with distinct indices, i.e., $i_1 \neq i_2 \ldots \neq i_n$. The former subset is exactly $N_A(S)$ while the size of the latter subset is at least $|S|$ due to the existence of a perfect n-gram matching in A. The bound (36) is used in the second inequality. Since $d_{\max}\big(A^{\odot n}\big) = d_{\max}(A)^n$, the proof of part 2 is also completed.

■

Remark 31 *The second result of above lemma is similar to the necessity argument of (Hall's) Theorem 32 for the existence of perfect matching in a bipartite graph, but generalized to the case of perfect n-gram matching and with additional krank condition.*

A.4 Auxiliary Lemma

Proof of Lemma 1: We show that if $G(Y, X; A)$ has a perfect n-gram matching, then $G(Y, X^{(n)}; A^{\odot n})$ has a perfect matching. The reverse can be also immediately shown by reversing the discussion and exploiting the additional condition stated in the lemma.

19. The Lemma 6 result is about the column rank of A itself, but here it is about the column rank of $A^{\odot n}$ for which the same analysis works. Note that the support of B_M (which is full column rank here) is within the support of $A^{\odot n}$ and therefore Lemma 6 can still be applied.
20. Note that many terms in this bound are ignored which leads to a loose bound that might be improved.

Let $E^{\odot n}$ denote the edge set of the bipartite graph $G(Y, X^{(n)}; A^{\odot n})$. Assume $G(Y, X; A)$ has a perfect n-gram matching $M \subseteq E$. For any $j \in Y$, let $N_M(j)$ denote the set of neighbors of vertex j according to edge set M. Since M is a perfect n-gram matching, $|N_M(j)| = n$ for all $j \in Y$. It can be immediately concluded from Definition 4 that sets $N_M(j)$ are all distinct, i.e., $N_M(j_1) \neq N_M(j_2)$ for any $j_1, j_2 \in Y, j_1 \neq j_2$. For any $j \in Y$, let $N'_M(j)$ denote an arbitrary ordered n-tuple generated from the elements of set $N_M(j)$. From the definition of n-gram matrix, we have $A^{\odot n}(N'_M(j), j) \neq 0$ for all $j \in Y$. Hence, $(j, N'_M(j)) \in E^{\odot n}$ for all $j \in Y$ which together with the fact that all $N'_M(j)$'s tuples are distinct, it results that $M^{\odot n} := \{(j, N'_M(j)) | j \in Y\} \subseteq E^{\odot n}$ is a perfect matching for $G(Y, X^{(n)}; A^{\odot n})$. ■

Lemma 6 *Consider matrix $C \in \mathbb{R}^{m \times r}$ which is generic. Let $\widetilde{C} \in \mathbb{R}^{m \times r}$ be such that $\text{Supp}(\widetilde{C}) \subseteq \text{Supp}(C)$ and the non-zero entries of $\widetilde{C}$ are the same as the corresponding non-zero entries of C. If $\widetilde{C}$ is full column rank, then C is also full column rank, almost surely.*

Proof: Since $\widetilde{C}$ is full column rank, there exists a $r \times r$ submatrix of $\widetilde{C}$, denoted by $\widetilde{C}_S$, with non-zero determinant, i.e., $\det(\widetilde{C}_S) \neq 0$. Let C_S denote the corresponding submatrix of C indexed by the same rows and columns as $\widetilde{C}_S$.
The determinant of C_S is a polynomial in the entries of C_S. Since $\widetilde{C}_S$ can be derived from C_S by keeping the corresponding non-zero entries, $\det(C_S)$ can be decomposed into two terms as

$$\det(C_S) = \det(\widetilde{C}_S) + f(C_S),$$

where the first term corresponds to the monomials for which all the variables (entries of C_S) are also in $\widetilde{C}_S$ and the second term corresponds to the monomials for which at least one variable is not in $\widetilde{C}_S$. The first term is non-zero as stated earlier. Since C is generic, the polynomial $f(C_S)$ is non-trivial and therefore its roots have Lebesgue measure zero. It implies that $\det(C_S) \neq 0$ with Lebesgue measure one (almost surely), and hence, it is full (column) rank. Thus, C is also full column rank, almost surely. ■

Finally, Theorem 9 is proved by combining the results of Theorem 27 and Lemma 5.
Proof of Theorem 9: Since conditions 2 and 3 hold and A is generic, Lemma 5 can be applied which results that rank condition 6 is satisfied almost surely and expansion condition 7 also holds. Therefore, all the required conditions for Theorem 27 are satisfied almost surely and this completes the proof. ■

Appendix B. Proof of Random Identifiability Result (Theorem 15)

We provide detailed proof of the steps stated in the proof sketch of random result in Section 5.2.

B.1 Proof of Existence of Perfect n-gram Matching and Kruskal Results

Restatement of Theorem 22 *Consider a random bipartite graph $G(Y, X; E)$ with $|Y| = q$ nodes on the left side and $|X| = p$ nodes on the right side, and each node $i \in Y$ is randomly connected to d_i different nodes in X. Let $d_{\min} := \min_{i \in Y} d_i$. Assume that it satisfies the size condition $q \leq \left(c\frac{p}{n}\right)^n$ (condition 4) for some constant $0 < c < 1$ and the degree condition $d_{\min} \geq \max\{1 + \beta \log p, \alpha \log p\}$ for some constants $\beta > \frac{n-1}{\log 1/c}, \alpha >$*

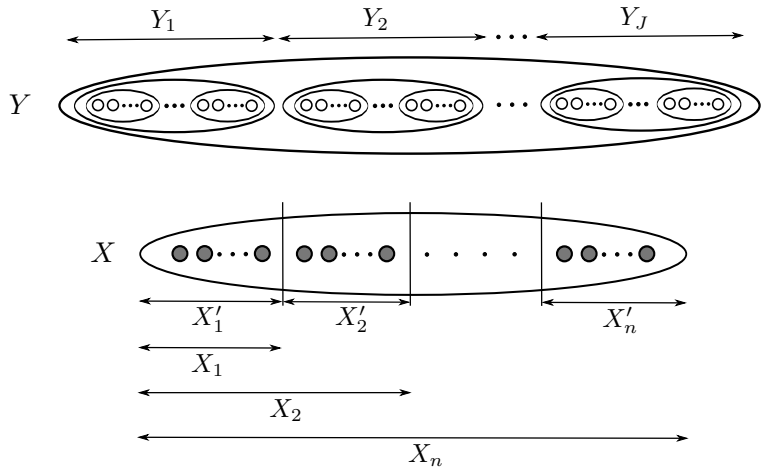


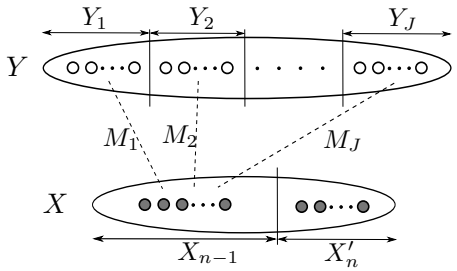
Figure 6: Partitioning of sets Y and X, proposed in the proof of Theorem 22. Set X is randomly (uniform) partitioned into n sets of (almost) equal size, denoted by $X'_l, l \in [n]$. Set Y is also randomly partitioned in a recursive manner. In each step, it is partitioned to $J = c\frac{p}{n} = O(p)$ number of sets. These smaller sets are again partitioned, recursively. This partitioning process is performed until reaching sets with size $O(p)$. The first two steps are shown in this figure.

$\max\{2n^2(\beta \log \frac{1}{c} + 1), 2\beta n\}$ *(lower bound in condition 5). Then, there exists a perfect (Y-saturating) n-gram matching in the random bipartite graph* $G(Y, X; E)$*, with probability at least* $1 - \gamma_1 p^{-\beta'}$ *for constants* $\beta' > 0$ *and* $\gamma_1 > 0$*, specified in* (5) *and* (6)*.*

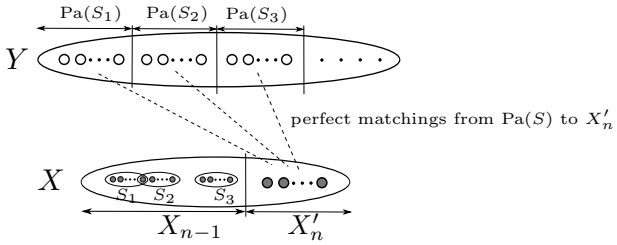
Proof of Theorem 22: Vertex sets X and Y are partitioned, described as follows (see Figure 6). Define $J := c\frac{p}{n}$. Partition set X uniformly at random into n sets of (almost) equal size[21], denoted by $X'_l, l \in [n]$. Define sets $X_l := \cup_{i=1}^{l} X'_i, l \in [n]$. Furthermore, partition set Y uniformly at random, hierarchically as follows. First, partition into J sets, each with size at most $\left(c\frac{p}{n}\right)^{n-1}$, and denote them by $Y_i, i \in [J]$. Next, partition each of these new smaller sets Y_i further into J sets, each with size at most $\left(c\frac{p}{n}\right)^{n-2}$. Do it iteratively up to $n-1$ steps, where at the end, set Y is partitioned into sets with size at most $c\frac{p}{n}$. The first two steps are shown in Figure 6.

Proof by induction: The existence of perfect n-gram matching from set Y to set X is proved by an induction argument. Consider one of intermediate sets in the hierarchical partitioning of Y with size $O(p^l)$ and its further partitioning into $J := c\frac{p}{n}$ sets, each with size $O(p^{l-1})$, for any $l \in \{2, \ldots, n\}$. In the induction step, it is shown that if there exists a perfect $(l-1)$-gram matching from each of these subsets of Y with size $O(p^{l-1})$ to X_{l-1}, then there exists a perfect l-gram matching from the original set with size $O(p^l)$ to set X_l. Specifically, in the last induction step, it is shown that if there exists a perfect $(n-1)$-gram matching from each set $Y_l, l \in [J]$, to set X_{n-1}, then there exists a perfect n-gram matching from Y to $X_n = X$.

21. By almost, we mean the maximum difference in the size of partitions is 1 which is always possible.



(a) Partitioning of sets Y and X proposed for the induction step.



(b) Partitioning of set Y through perfect $(n-1)$-gram matchings $M_i, i \in [J]$.

Figure 7: Auxiliary figures for proof of induction step. (a) Partitioning of sets Y and X proposed in the proof, where set Y is partitioned to $J := c\frac{p}{n}$ partitions $Y_1, \ldots, Y_J$ with (almost) equal size, for some constant $c < 1$. In addition, set X is partitioned to two partitions X_{n-1} and X'_n with sizes $|X_{n-1}| = \frac{n-1}{n}p$ and $|X'_n| = \frac{p}{n}$. The perfect $(n-1)$-gram matchings $M_i, i \in [J]$, through bipartite graphs $G_i(Y_i, X_{n-1}; E_i), i \in [J]$, are also highlighted in the figure. (b) Set Y is partitioned to subsets $\mathrm{Pa}(S), S \in P_{n-1}(X_{n-1})$, which is generated through perfect $(n-1)$-gram matchings $M_i, i \in [J]$. S_1, S_2 and S_3 are three different sets in $P_{n-1}(X_{n-1})$ shown as samples. In addition, the perfect matchings from $\mathrm{Pa}(S), S \in P_{n-1}(X_{n-1})$, to X'_n, proposed in the proof, are also highlighted in the figure.

Base case of induction: The base case of induction argument holds as follows. By applying Lemma 8 and Lemma 7, there exists a perfect matching from each partition in Y with size at most $c\frac{p}{n} = O(p)$ to set X_1, **whp**.

Induction step: Consider J different bipartite graphs $G_i(Y_i, X_{n-1}; E_i), i \in [J]$, by considering sets Y_i and X_{n-1} and the corresponding subset of edges $E_i \subset E$ incident to them. See Figure 7a. The induction step is to show that if each of the corresponding J bipartite graphs $G_i(Y_i, X_{n-1}; E_i), i \in [J]$, has a perfect $(n-1)$-gram matching, then **whp**, the original bipartite graph $G(Y, X; E)$ has a perfect n-gram matching.

Let us denote the corresponding perfect $(n-1)$-gram matching of $G_i(Y_i, X_{n-1}; E_i)$ by M_i. Furthermore, the set of all subsets of X_{n-1} with cardinality $n-1$ are denoted by $P_{n-1}(X_{n-1})$, i.e., $P_{n-1}(X_{n-1})$ includes the sets with $(n-1)$ elements in the power set [22] of X_{n-1}. For each set $S \in P_{n-1}(X_{n-1})$, take the set of all nodes in Y which are connected to all members of S according to the union of matchings $\cup_{i=1}^{J} M_i$. Call this set the parents of S, denoted by $\mathrm{Pa}(S)$. According to the definition of perfect $(n-1)$-gram matching, there is at most one node in each set Y_i which is connected to all members of S through the matching M_i and therefore, $|\mathrm{Pa}(S)| \leq J = c\frac{p}{n}$. In addition, note that sets $\mathrm{Pa}(S)$ impose a partitioning on set Y, i.e., each node $j \in Y$ is exactly included in one set $\mathrm{Pa}(S)$ for some $S \in P_{n-1}(X_{n-1})$. This is because of the perfect $(n-1)$-gram matchings considered for sets $Y_i, i \in [J]$.

Now, a perfect n-gram matching for the original bipartite graph is constructed as follows. For any $S \in P_{n-1}(X_{n-1})$, consider the set of parents $\mathrm{Pa}(S)$. Create the bipartite graph $G_S(\mathrm{Pa}(S), X'_n; E_S)$, where $E_S \subset E$ is the subset of edges incident to partitions $\mathrm{Pa}(S) \subset Y$

22. The power set of any set S is the set of all subsets of S.

and $X'_n \subset X$. Denote by d_S the minimum degree of nodes in set $\mathrm{Pa}(S)$ in the bipartite graph $G_S(\mathrm{Pa}(S), X'_n; E_S)$. Applying Lemma 8, we have

$$\begin{aligned}\Pr[d_S \geq 1 + \beta \log(p/n)] &\geq 1 - J \exp\left(-\frac{2}{n^2}\frac{(d_{\min} - \beta n \log(p/n))^2}{d_{\min}}\right) \\ &\geq 1 - \frac{c}{n} p^{-\beta \log 1/c} = 1 - O(p^{-\beta \log 1/c}),\end{aligned} \tag{38}$$

where $\beta \log 1/c > n-1$, and the last inequality is concluded from the degree bound $d_{\min} \geq \alpha \log p$. Furthermore, we have $|\mathrm{Pa}(S)| \leq c\frac{p}{n} = c|X'_n|$. Now, we can apply Lemma 7 concluding that there exists a perfect matching from $\mathrm{Pa}(S)$ to X'_n within the bipartite graph $G_S(\mathrm{Pa}(S), X'_n; E_S)$, with probability at least $1 - O(p^{-\beta \log 1/c})$. Refer to Figure 7b for a schematic picture. The edges of this perfect matching are combined with the corresponding edges of the existing perfect $(n-1)$-gram matchings $M_i, i \in [J]$, to provide n incident edges to each node $i \in \mathrm{Pa}(S)$. It is easy to see that this provides a perfect n-gram matching from $\mathrm{Pa}(S)$ to X.

We perform the same steps for all sets $S \in P_{n-1}(X_{n-1})$ to obtain a perfect n-gram matching from any $\mathrm{Pa}(S), S \in P_{n-1}(X_{n-1})$, to X. Finally, according to this construction, the union of all of these matchings is a perfect n-gram matching from $\cup_{S \in P_{n-1}(X_{n-1})} \mathrm{Pa}(S) = Y$ to X. This finishes the proof of induction step. Note that here we analyzed the last induction step where the existence of perfect n-gram matching is concluded from the existence of corresponding perfect $(n-1)$-gram matchings. The earlier induction steps, where the existence of perfect l-gram matching is concluded from the existence of corresponding perfect $(l-1)$-gram matchings for any $l \in \{2, \ldots, n\}$, can be similarly proven.

Probability rate: We now provide the probability rate of the above events. Let $N_l^{(\mathrm{hp})}, l \in [n]$, denote the total number of times that perfect matching result of Lemma 7 is used in step l in order to ensure that there exists a perfect l-gram matching from corresponding partitions of Y to set X_l, **whp**. Let $N^{(\mathrm{hp})} = \sum_{l \in [n]} N_l^{(\mathrm{hp})}$. As earlier, let $P_{l-1}\big(X_{l-1}\big)$ denote the set of all subsets of X_{l-1} with cardinality $l-1$. We have

$$\big|P_{l-1}\big(X_{l-1}\big)\big| = \binom{\big|X_{l-1}\big|}{l-1} = \binom{\frac{l-1}{n}p}{l-1}, \quad l \in \{2, \ldots, n\}.$$

According to the construction method of l-gram matching from $(l-1)$-gram matchings, proposed in the induction step, $\big|P_{l-1}\big(X_{l-1}\big)\big|$ is the number of times Lemma 7 is used in order to ensure that there exists a perfect l-gram matching for each partition on the Y side. Since at most J^{n-l} number of such l-gram matchings are proposed in step l, the number $N_l^{(\mathrm{hp})}$ can be bounded as

$$N_l^{(\mathrm{hp})} \leq J^{n-l} \, \big|P_{l-1}\big(X_{l-1}\big)\big| = J^{n-l}\binom{\frac{l-1}{n}p}{l-1}, \quad l \in \{2, \ldots, n\}. \tag{39}$$

Since in the first step, $N_1^{(\mathrm{hp})} = J^{n-1}$ number of perfect matchings needs to exist in the above discussion, we have

$$N^{(\mathrm{hp})} = J^{n-1} + \sum_{l=2}^{n} N_l^{(\mathrm{hp})}$$

$$\leq J^{n-1} + \sum_{l=2}^{n} J^{n-l} \binom{\frac{l-1}{n}p}{l-1}$$
$$\leq \Big(c\frac{p}{n}\Big)^{n-1} + \sum_{l=2}^{n} \Big(c\frac{p}{n}\Big)^{n-l} \Big(e\frac{p}{n}\Big)^{l-1}$$
$$\leq n\Big(e\frac{p}{n}\Big)^{n-1} = O(p^{n-1}),$$

where inequality (39) is used in the first inequality and $J := c\frac{p}{n}$ and inequality $\binom{n}{k} \leq \big(e\frac{n}{k}\big)^k$ are exploited in the second inequality.
Since the result of Lemma 7 holds with probability at least $1 - O(p^{-\beta \log 1/c})$ and it is assumed that $\beta \log 1/c > n-1$, by applying union bound, we have the existence of perfect n-gram matching with probability at least $1 - O(p^{-\beta'})$, for $\beta' = \beta \log \frac{1}{c} - (n-1) > 0$.
Furthermore, note that the degree concentration bound in (38) is also used $O(p^{n-1})$ times. Since the bound in (38) holds with probability at least $1 - O(p^{-\beta \log 1/c})$ and it is assumed that $\beta \log 1/c > n-1$, this also reduces to the same probability rate.
The coefficient of the above polynomial probability rate is also explicitly computed, saying that the perfect n-gram matching exists with probability at least $1 - \gamma_1 p^{-\beta'}$, with

$$\gamma_1 = e^{n-1}\Big(\frac{c}{n^{n-1}} + \frac{e^2}{1-\delta_1} n^{\beta'+1}\Big),$$

where δ_1 is a constant satisfying $e^2\Big(\frac{p}{n}\Big)^{-\beta \log 1/c} < \delta_1 < 1$. ■

Proof of Theorem 24: Let $G(Y, X; A)$ denote the corresponding bipartite graph to matrix A where node sets $Y = [q]$ and $X = [p]$ index the columns and rows of A respectively. Therefore, $|Y| = q$ and $|X| = p$. Fix some $S \subseteq Y$ such that $|S| \leq p$. Then

$$\begin{aligned}
\Pr(|N(S)| \leq |S|) &\leq \sum_{\substack{T \subseteq X: \\ |T|=|S|}} \Pr(N(S) \subseteq T) \\
&= \sum_{\substack{T \subseteq X: \\ |T|=|S|}} \prod_{i \in S} \binom{|S|}{d_i} \Big/ \binom{p}{d_i} \\
&\leq \sum_{\substack{T \subseteq X: \\ |T|=|S|}} \prod_{i \in S} \Big(\frac{|S|}{p}\Big)^{d_i} \\
&\leq \sum_{\substack{T \subseteq X: \\ |T|=|S|}} \prod_{i \in S} \Big(\frac{|S|}{p}\Big)^{d_{\min}} \\
&= \binom{p}{|S|} \Big(\frac{|S|}{p}\Big)^{d_{\min}|S|}, \qquad (40)
\end{aligned}$$

where the bound $\binom{|S|}{d_i} \Big/ \binom{p}{d_i} \leq \Big(\frac{|S|}{p}\Big)^{d_i}$ is used in the second inequality, and the last inequality is concluded from the fact that $\frac{|S|}{p} \leq 1$.

Let $\mathcal{E}$ denote the event that for any subset $S \subseteq Y$ with $|S| \leq r$, we have $|N(S)| \geq |S|$, i.e.,

$$\mathcal{E} := \text{“}\forall S \subseteq Y \wedge 1 \leq |S| \leq r : |N(S)| \geq |S|\text{”}.$$

Then, by the union bound and inequality (40), we have

$$\begin{aligned}
\Pr(\mathcal{E}^c) = \Pr(\exists S \subseteq Y \,\text{s.t.}\, 1 \leq |S| \leq r \wedge |N(S)| < |S|) &\leq \sum_{s=1}^{r} \binom{q}{s}\binom{p}{s}\left(\frac{s}{p}\right)^{d_{\min} s} \\
&\leq \sum_{s=1}^{r} \left(e\frac{q}{s}\right)^{s}\left(e\frac{p}{s}\right)^{s}\left(\frac{s}{p}\right)^{d_{\min} s} \\
&\leq \sum_{s=1}^{r} \left(\frac{e^2 q r^{d_{\min}-2}}{p^{d_{\min}-1}}\right)^{s},
\end{aligned}$$

where the bound $\binom{n}{k} \leq \left(e\frac{n}{k}\right)^k$ is used in the second inequality. For $r = cp$, the above inequality reduces to

$$\begin{aligned}
\Pr(\mathcal{E}^c) &\leq \sum_{s=1}^{r} \left(e^2 c^{d_{\min}-2}\frac{q}{p}\right)^{s} \\
&\leq \sum_{s=1}^{r} \left(e^2 c' c^{d_{\min}-1} p^{n-1}\right)^{s} \\
&\leq \sum_{s=1}^{r} \left(e^2 c' c^{\beta \log p} p^{n-1}\right)^{s} \\
&= \sum_{s=1}^{r} \left(e^2 c' p^{n-1-\beta \log 1/c}\right)^{s} \\
&\leq \frac{e^2 c'}{p^{\beta'} - e^2 c'} = O(p^{-\beta'}), \quad \text{for } \beta' = \beta \log \frac{1}{c} - (n-1) > 0,
\end{aligned}$$

where the size condition assumed in the theorem is used in the second inequality with $c' := \frac{1}{c}\left(\frac{c}{n}\right)^n$, and the degree condition is exploited in the third inequality. The last inequality is concluded from the geometric series sum formula for large enough p.
Then, Lemma 9 can be applied concluding that $\text{krank}(A) \geq r = cp$, with probability at least $1 - \gamma_2 p^{-\beta'}$ for constants $\beta' = \beta \log \frac{1}{c} - (n-1) > 0$ and $\gamma_2 > 0$ as

$$\gamma_2 = \frac{c^{n-1} e^2}{n^n (1 - \delta_2)},$$

where δ_2 is a constant satisfying $c' e^2 p^{-\beta'} < \delta_2 < 1$. ■

Proof of Remark 23: Consider a random bipartite graph $G(Y, X; E)$ where for each node $i \in X$:

1. Neighbors $N(i) \subseteq X$ are picked uniformly at random among all size d subsets of X.
2. Matching $M(i) \subseteq N(i)$ is picked uniformly at random among all size n subsets of $N(i)$.

Note that as long as $n \leq d$, the distribution of $M(i)$ is uniform over all size n subsets of X. Fix some pair $i, i' \in Y$. Then

$$\Pr(M(i) = M(i')) = \binom{|X|}{n}^{-1}.$$

By the union bound,

$$\Pr\Big(\exists i, i' \in Y, i \neq i' \,\text{s.t.}\, M(i) = M(i')\Big) \leq \binom{|Y|}{2}\binom{|X|}{n}^{-1},$$

which is $\Theta(|Y|^2/|X|^n)$ when n is constant. Therefore, if $d \geq n$ and the size constraint $|Y| = O(|X|^s)$ for some $s < \frac{n}{2}$ is satisfied, then **whp**, there is no pair of nodes in set Y with the same random n-gram matching. This concludes that the random bipartite graph has a perfect n-gram matching **whp**, under these size and degree conditions.

■

B.2 Auxiliary Lemmata

Lemma 7 (Existence of perfect matching for random bipartite graphs) *Consider a random bipartite graph $G(W, Z; E)$ with $|W| = w$ nodes on the left side and $|Z| = z$ on the right side, and each node $i \in W$ is randomly connected to d_i different nodes in set Z. Let $d_w := \min_{i \in W} d_i$. Assume that it satisfies the size condition $w \leq cz$ for some constant $0 < c < 1$ and the degree condition $d_w \geq 1 + \beta \log z$ for some constant $\beta > 0$. Then, there exists a perfect matching in the random bipartite graph $G(W, Z; E)$ with probability at least $1 - O(z^{-\beta \log 1/c})$ where $\beta \log \frac{1}{c} > 0$.*

Proof: From Hall's theorem (Theorem 32), the existence of perfect matching for a bipartite graph is equivalent to occurrence of the following event

$$\widetilde{\mathcal{E}} := \text{“}\forall S \subseteq W : |N(S)| \geq |S|\text{”}.$$

Similar to the analysis in the proof of Theorem 24, applying the union bound we have

$$\begin{aligned}
\Pr\big(\widetilde{\mathcal{E}}^c\big) = \Pr(\exists S \subseteq W \,\text{s.t.}\, |N(S)| < |S|) &\leq \sum_{s=1}^{w} \binom{w}{s}\binom{z}{s}\left(\frac{s}{z}\right)^{d_w s} \\
&\leq \sum_{s=1}^{w} \left(e\frac{w}{s}\right)^{s}\left(e\frac{z}{s}\right)^{s}\left(\frac{s}{z}\right)^{d_w s} \\
&\leq \sum_{s=1}^{w} \left(\frac{e^2 w^{d_w - 1}}{z^{d_w - 1}}\right)^{s} \\
&\leq \sum_{s=1}^{w} \Big(e^2 c^{d_w - 1}\Big)^{s},
\end{aligned}$$

where the bound $\binom{n}{k} \leq \big(e\frac{n}{k}\big)^k$ is used in the second inequality. From the assumed lower bound on the degree d_w and the fact that $0 < c < 1$, we have

$$\Pr\big(\widetilde{\mathcal{E}}^c\big) \leq \sum_{s=1}^{w} \Big(e^2 c^{\beta \log z}\Big)^{s} = \sum_{s=1}^{w} \Big(e^2 z^{\beta \log c}\Big)^{s} \leq \frac{e^2}{z^{\beta \log \frac{1}{c}} - e^2} \leq \frac{e^2}{1 - \delta_1} z^{-\beta \log 1/c},$$

where the second inequality is concluded from the geometric series sum formula for large enough z, and δ_1 is a constant satisfying $e^2 z^{-\beta \log 1/c} < \delta_1 < 1$. ■

Lemma 8 (Degree concentration bound) *Consider a random bipartite graph $G(Y, X; E)$ with $|Y| = q$ and $|X| = p$, where each node $i \in Y$ is randomly connected to d_i different nodes in set X. Let $Y' \subset Y$ be any subset*[23] *of nodes in Y with size $|Y'| = q'$ and $X' \subset X$ be a random (uniformly chosen) subset of nodes in X with size $|X'| = p'$. Create the new bipartite graph $G(Y', X'; E')$ where edge set $E' \subset E$ is the subset of edges in E incident to Y' and X'. Denote the degree of each node $i \in Y'$ within this new bipartite graph by d'_i. Let $d_{\min} := \min_{i \in Y} d_i$ and $d'_{\min} := \min_{i \in Y'} d'_i$. Then, if $d_{\min} > r\frac{p}{p'}$ for a non-negative integer r, we have*

$$\Pr[d'_{\min} \geq r+1] \geq 1 - q' \exp\left(-2(p'/p)^2 \frac{(d_{\min} - (p/p')r)^2}{d_{\min}}\right).$$

Proof: For any $i \in Y'$, we have

$$\Pr[d'_i \leq r] = \sum_{j=0}^{r} \binom{p'}{j} \binom{p-p'}{d_i - j} \Big/ \binom{p}{d_i},$$

where the inner term of summation is a hypergeometric distribution with parameters p (population size), p' (number of success states in the population), d_i (number of draws) and j is the hypergeometric random variable denoting number of successes. The following tail bound for the hypergeometric distribution is provided (Chvátal, 1979; Skala)

$$\Pr[d'_i \leq r] \leq \exp(-2t_i^2 d_i),$$

for $t_i > 0$ given by $r = \big(\frac{p'}{p} - t_i\big)d_i$. Note that assumption $d_{\min} > \frac{p}{p'}r$ in the lemma is equivalent to having $t_i > 0, i \in Y$. Considering the minimum degree, for any $i \in Y'$, we have

$$\Pr[d'_i \leq r] \leq \exp(-2t^2 d_{\min}),$$

for $t > 0$ given by $r = \big(\frac{p'}{p} - t\big)d_{\min}$. Substituting t from this equation gives the following bound

$$\Pr[d'_i \leq r] \leq \exp\left(-2(p'/p)^2 \frac{(d_{\min} - (p/p')r)^2}{d_{\min}}\right). \tag{41}$$

Finally, applying the union bound, we can prove the result as follows

$$\begin{aligned}
\Pr[d'_{\min} \geq r+1] &= \Pr[\cap_{i=1}^{q'} \{d'_i \geq r+1\}] \\
&\geq 1 - \sum_{i=1}^{q'} \Pr[d'_i \leq r] \\
&\geq 1 - \sum_{i=1}^{q'} \exp\left(-2(p'/p)^2 \frac{(d_{\min} - (p/p')r)^2}{d_{\min}}\right)
\end{aligned}$$

23. Note that Y' need not to be uniformly chosen and the result is valid for any subset of nodes $Y' \subset Y$.

$$=1-q'\exp\bigg(-2(p'/p)^2\frac{(d_{\min}-(p/p')r)^2}{d_{\min}}\bigg),$$

where the union bound is applied in the first inequality and the second inequality is concluded from (41). ■

A lower bound on the Kruskal rank of matrix A based on a sufficient relaxed expansion property on A is provided in the following lemma which might have independent interest.

Lemma 9 *If A is generic and the bipartite graph $G(Y,X;A)$ satisfies the relaxed[24] expansion property $|N(S)|\geq|S|$ for any subset $S\subseteq Y$ with $|S|\leq r$, then* $\text{krank}(A)\geq r$, *almost surely.*

Before proposing the proof, we state the marriage or Hall's theorem which gives an equivalent condition for having a perfect matching in a bipartite graph.

Theorem 32 (Hall's theorem, (Hall, 1935)) *A bipartite graph $G(Y,X;E)$ has Y-saturating matching if and only if for every subset $S\subseteq Y$, the size of the neighbors of S is at least as large as S, i.e., $|N(S)|\geq|S|$.*

Proof of Lemma 9: Denote the submatrix $A_{N(S),S}$ by $\widetilde{A}_S$, i.e., $\widetilde{A}_S:=A_{N(S),S}$. Exploiting marriage or Hall's theorem, it is concluded that the bipartite graph $G(S,N(S);\widetilde{A}_S)$ has a perfect matching M_S for any subset $S\subseteq Y$ such that $|S|\leq r$. Denote by $\widetilde{A}_{M_S}$ the corresponding matrix to this perfect matching edge set M_S, i.e., $\widetilde{A}_{M_S}$ keeps the non-zero entries of $\widetilde{A}_S$ on edge set M_S and everywhere else, it is zero. Note that the support of $\widetilde{A}_{M_S}$ is within the support of $\widetilde{A}_S$. According to the definition of perfect matching, the matrix $\widetilde{A}_{M_S}$ is full column rank. From Lemma 6, it is concluded that $\widetilde{A}_S$ is also full column rank almost surely. This is true for any $\widetilde{A}_S$ with $S\subseteq Y$ and $|S|\leq r$, which directly results that $\text{krank}(A)\geq r$, almost surely. ■

Finally, Theorem 15 is proved by exploiting the random results on the existence of perfect n-gram matching and Kruskal rank, provided in Theorems 22 and 24.

Proof of Theorem 15: We claim that if random conditions 4 and 5 are satisfied, then deterministic conditions 2 and 3 hold **whp**. Then Theorem 9 can be applied and the proof is done.

From size and degree conditions, Theorem 22 can be applied, which implies that the perfect n-gram matching condition 2 is satisfied with probability at least $1-\gamma_1 p^{-\beta'}$ for $\beta'=\beta\log\frac{1}{c}-(n-1)>0$. The conditions required for Theorem 24 also hold and by applying this theorem we have the bound $\text{krank}(A)\geq cp$, with probability at least$1-\gamma_2 p^{-\beta'}$. Combining this inequality with the upper bound on degree d in condition 5, we conclude that krank condition 3 is also satisfied **whp**. Hence, all the conditions required for Theorem 9 are satisfied with probability at least $1-\gamma p^{-\beta'}$, where

$$\gamma=\gamma_1+\gamma_2=e^{n-1}\Big(\frac{c}{n^{n-1}}+\frac{e^2}{1-\delta_1}n^{\beta'+1}\Big)+\frac{c^{n-1}e^2}{n^n(1-\delta_2)},$$

and this completes the proof. ■

Finally, Corollary 19 can be also proved by showing that the size and degree conditions satisfy the full column rank condition required in Corollary 13. This is proved in Lemma 7.

24. There is no $d_{\max}$ term in contrast to the expansion property proposed in condition 7.

Appendix C. Relationship to CP Decomposition Uniqueness Results

In this section, we provide a more detailed comparison with some uniqueness results of overcomplete CP decomposition. Here, the following CP decomposition for the third order tensor $T \in \mathbb{R}^{p\times s\times q}$ is considered,

$$T = \sum_{i=1}^{r} a_i \circ b_i \circ c_i, \tag{42}$$

where $A = [a_1|\dots|a_r] \in \mathbb{R}^{p\times r}, B = [b_1|\dots|b_r] \in \mathbb{R}^{s\times r}$ and $C = [c_1|\dots|c_r] \in \mathbb{R}^{q\times r}$.
The most important and general uniqueness result of CP, called Kruskal's condition, is provided in Kruskal (1977), where it is guaranteed that the above CP decomposition is unique if

$$\mathrm{krank}(A) + \mathrm{krank}(B) + \mathrm{krank}(C) \geq 2r + 2.$$

Since then, several works have analyzed the uniqueness of CP decomposition. One set of works assume that one of the components, say C, is full column rank (Lathauwer, 2006; Jiang and Sidiropoulos, 2004). It is shown in Lathauwer (2006), for generic (fully dense) components A, B and C, if $r \leq q$ and $r(r-1) \leq p(p-1)s(s-1)/2$, then the CP decomposition in (42) is generically unique.
Now, we demonstrate how this CP uniqueness result can be adapted to our setting. First, consider the matrix $M \in \mathbb{R}^{ps\times q}$ which is obtained by stacking the entries of T as

$$M_{(i-1)s+j,k} = T_{ijk}.$$

Then, we have

$$M = (A \odot B)C^\top. \tag{43}$$

On the other hand, for the 2-persistent topic model with 4 words ($n = 2, m = 2$), the moment can be written as

$$M_4^{(2)}(x) = (A \odot A)\mathbb{E}\big[hh^\top\big](A \odot A)^\top,$$

for $A \in \mathbb{R}^{p\times q}$. The following matrix has the same column span of $M_4^{(2)}(x)$,

$$M' = (A \odot A)C'^\top,$$

for some full rank matrix $C' \in \mathbb{R}^{q\times q}$. Our random identifiability result in Theorem 15 provides the uniqueness of A and C', given M', under the size condition $q \leq \left(c\frac{p}{2}\right)^2$ and the additional degree condition 5. Note that as discussed in the previous section, this identifiability argument is the same as the unique decomposition of the corresponding tensor.
Thus, in equation (43), by setting $A = B$ and a full rank square matrix C, we obtain the 2-persistent topic model, under consideration in this paper. Thus, the identifiability results of Lathauwer (2006) are applicable to our setting, if we assume generic (i.e. fully dense) matrix A. However, we incorporate a sparse matrix A, and therefore, require different techniques to provide identifiability results. We note that the size bound specified in Lathauwer (2006) is comparable to the size bound derived in this paper (for random structured matrices), but we have additional degree considerations for identifiability. Analyzing the regime where the uniqueness conditions of Lathauwer (2006) are satisfied under sparsity constraints is an interesting question for future investigation.

References

Elizabeth S. Allman, John A. Rhodes, and Amelia Taylor. A semialgebraic description of the general markov model on phylogenetic trees. *Arxiv preprint arXiv:1212.1200*, Dec. 2012.

E.S. Allman, C. Matias, and J.A. Rhodes. Identifiability of parameters in latent structure models with many observed variables. *The Annals of Statistics*, 37(6A):3099–3132, 2009.

A. Anandkumar, D. P. Foster, D. Hsu, S. M. Kakade, and Y. K. Liu. A Spectral Algorithm for Latent Dirichlet Allocation. In *Proc. of Neural Information Processing (NIPS)*, Dec. 2012a.

A. Anandkumar, R. Ge, D. Hsu, S. M. Kakade, and M. Telgarsky. Tensor Methods for Learning Latent Variable Models. *Under Review. J. of Machine Learning. Available at arXiv:1210.7559*, Oct. 2012b.

A. Anandkumar, D. Hsu, A. Javanmard, and S. M. Kakade. Learning Linear Bayesian Networks with Latent Variables. *ArXiv e-prints*, September 2012.

A. Anandkumar, D. Hsu, and S.M. Kakade. A Method of Moments for Mixture Models and Hidden Markov Models. In *Proc. of Conf. on Learning Theory*, June 2012.

A. Anandkumar, R. Ge, D. Hsu, and S. M. Kakade. A Tensor Spectral Approach to Learning Mixed Membership Community Models. In *Conference on Learning Theory (COLT)*, June 2013.

Saneev Arora, Rong Ge, and Ankur Moitra. Learning topic models—going beyond svd. In *Symposium on Theory of Computing*, 2012a.

Sanjeev Arora, Rong Ge, Yoni Halpern, David M. Mimno, Ankur Moitra, David Sontag, Yichen Wu, and Michael Zhu. A practical algorithm for topic modeling with provable guarantees. *ArXiv 1212.4777*, 2012b.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Unsupervised feature learning and deep learning: A review and new perspectives. *arXiv preprint arXiv:1206.5538*, 2012.

A. Bhaskara, M. Charikar, and A. Vijayaraghavan. Uniqueness of Tensor Decompositions with Applications to Polynomial Identifiability. *ArXiv 1304.8087*, April 2013.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

Cristiano Bocci, Luca Chiantini, and Giorgio Ottaviani. Refined methods for the identifiability of tensors. *arXiv preprint arXiv:1303.6915*, 2013.

J.T. Chang. Full reconstruction of markov models on evolutionary trees: identifiability and consistency. *Mathematical Biosciences*, 137(1):51–73, 1996.

Luca Chiantini and Giorgio Ottaviani. On generic identifiability of 3-tensors of small rank. *SIAM Journal on Matrix Analysis and Applications*, 33(3):1018–1037, 2012.

Luca Chiantini, Massimiliano Mella, and Giorgio Ottaviani. One example of general unidentifiable tensors. *arXiv preprint arXiv:1303.6914*, 2013.

V. Chvátal. The tail of the hypergeometric distribution. *Discrete Mathematics*, 25(3): 285–287, 1979.

Adam Coates, Honglak Lee, and Andrew Y. Ng. An analysis of single-layer networks in unsupervised feature learning. *Journal of Machine Learning Research - Proceedings Track*, 15:215–223, 2011.

L. De Lathauwer, J. Castaing, and J.-F Cardoso. Fourth-order cumulant-based blind identification of underdetermined mixtures. *IEEE Tran. on Signal Processing*, 55:2965–2973, June 2007.

Li Deng and Dong Yu. *Deep Learning for Signal and Information Processing*. NOW Publishers, 2013.

Silvia Gandy, Benjamin Recht, and Isao Yamada. Tensor completion and low-n-rank tensor recovery via convex optimization. *Inverse Problems*, 27(2):025010, 2011.

G.H. Golub and C.F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, Maryland, 2012.

Navin Goyal, Santosh Vempala, and Ying Xiao. Fourier pca. *ArXiv 1306.5825*, 2013.

Philip Hall. On representatives of subsets. *J. London Math. Soc.*, 10(1):26–30, 1935.

Christopher J Hillar and Friedrich T Sommer. Ramsey theory reveals the conditions when sparse coding on subsampled data is unique. *arXiv preprint arXiv:1106.3616*, 2011.

Piotr Indyk and Ilya Razenshteyn. On model-based RIP-1 matrices. *CoRR*, abs/1304.3604, 2013. URL `http://arxiv.org/abs/1304.3604`.

Tao Jiang and Nicholas D Sidiropoulos. Kruskal's permutation lemma and the identification of candecomp/parafac and bilinear models with constant modulus constraints. *Signal Processing, IEEE Transactions on*, 52(9):2625–2636, 2004.

M. Amin Khajehnejad, Alexandros G. Dimakis, Weiyu Xu, and Babak Hassibi. Sparse recovery of nonnegative signals with minimal expansion. *IEEE Transactions on Signal Processing*, 59(1):196–208, 2011.

Tamara Kolda and Brett Bader. Tensor decompositions and applications. *SIREV*, 51(3): 455–500, 2009.

Kenneth Kreutz-Delgado, Joseph F. Murray, Bhaskar D. Rao, Kjersti Engan, Te-Won Lee, and Terrence J. Sejnowski. Dictionary learning algorithms for sparse representation. *Neural Computation*, 15:349–396, February 2003.

J.B. Kruskal. More factors than subjects, tests and treatments: an indeterminacy theorem for canonical decomposition and individual differences scaling. *Psychometrika*, 41(3): 281–293, 1976.

J.B. Kruskal. Three-way arrays: Rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics. *Linear algebra and its applications*, 18(2):95–138, 1977.

Joseph M Landsberg. *Tensors: Geometry and applications*, volume 128. American Mathematical Soc., 2012.

Lieven De Lathauwer. A Link between the Canonical Decomposition in Multilinear Algebra and Simultaneous Matrix Diagonalization. *SIAM J. Matrix Analysis and Applications*, 28(3):642–666, 2006.

Quoc V. Le, Alexandre Karpenko, Jiquan Ngiam, and Andrew Y. Ng. ICA with Reconstruction Cost for Efficient Overcomplete Feature Learning. In *NIPS*, pages 1017–1025, 2011.

Michael S. Lewicki, Terrence J. Sejnowski, and Howard Hughes. Learning overcomplete representations. *Neural Computation*, 12:337–365, 1998.

Andreas Maurer, Massimiliano Pontil, and Bernardino Romera-Paredes. Sparse coding for multitask and transfer learning. *ArxXiv preprint*, abs/1209.0738, 2012.

Nishant A. Mehta and Alexander G. Gray. Sparsity-based generalization bounds for predictive sparse coding. In *Proc. of the Intl. Conf. on Machine Learning (ICML)*, Atlanta, USA, June 2013.

E. Mossel and S. Roch. Learning nonsingular phylogenies and hidden markov models. *The Annals of Applied Probability*, 16(2):583–614, 2006.

XuanLong Nguyen. Posterior contraction of the population polytope in finite admixture models. *arXiv preprint arXiv:1206.0068*, 2012.

J. K. Pritchard, M. Stephens, and P. Donnelly. Inference of population structure using multilocus genotype data. *Genetics*, 155:945–959, 2000.

Yuval Rabani, Leonard Schulman, and Chaitanya Swamy. Learning mixtures of arbitrary distributions over large discrete domains. *arXiv preprint arXiv:1212.1527*, 2012.

B. Rao and K. Kreutz-Delgado. An affine scaling methodology for best basis selection. *IEEE Tran. Signal Processing*, 47:187–200, January 1999.

Nicholas D. Sidiropoulos and Rasmus Bro. On the uniqueness of multilinear decomposition of N-way arrays. *Journal of Chemometrics*, 14(3):229–239, 2000.

Matthew Skala. Hypergeometric tail inequalities: ending the insanity. `http://ansuz.sooke.bc.ca/professional/hypergeometric.pdf`.

Daniel A. Spielman, Huan Wang, and John Wright. Exact recovery of sparsely-used dictionaries. *ArxXiv preprint*, abs/1206.5882, 2012a.

Daniel A Spielman, Huan Wang, and John Wright. Exact recovery of sparsely-used dictionaries. In *Proc. of Conf. on Learning Theory*, 2012b.

Alwin Stegeman, Jos M.F. Ten Berge, and Lieven De Lathauwer. Sufficient conditions for uniqueness in candecomp/parafac and indscal with random component matrices. *Psychometrika*, 71(2):219–229, June 2006.

André Uschmajew. Local convergence of the alternating least squares algorithm for canonical tensor approximation. *SIAM Journal on Matrix Analysis and Applications*, 33(2): 639–652, 2012.

Journal of Machine Learning Research 16 (2015) 2695-2724 Submitted 4/13; Revised 12/14; Published 12/15

Absent Data Generating Classifier for Imbalanced Class Sizes

Arash Pourhabib ARASH.POURHABIB@OKSTATE.EDU
School of Industrial Engineering and Management
Oklahoma State University
322 Engineering North, Stillwater, Oklahoma 74078-5016, USA

Bani K. Mallick BMALLICK@STAT.TAMU.EDU
Department of Statistics
Texas A&M University
3143 TAMU, College Station, TX 77843-3143, USA

Yu Ding YUDING@IEMAIL.TAMU.EDU
Department of Industrial and Systems Engineering
Texas A&M University
3131 TAMU, College Station, TX 77843-3131, USA

Editor: Russ Greiner

Abstract

We propose an algorithm for two-class classification problems when the training data are imbalanced. This means the number of training instances in one of the classes is so low that the conventional classification algorithms become ineffective in detecting the minority class. We present a modification of the kernel Fisher discriminant analysis such that the imbalanced nature of the problem is explicitly addressed in the new algorithm formulation. The new algorithm exploits the properties of the existing minority points to learn the effects of other minority data points, had they actually existed. The algorithm proceeds iteratively by employing the learned properties and conditional sampling in such a way that it generates sufficient artificial data points for the minority set, thus enhancing the detection probability of the minority class. Implementing the proposed method on a number of simulated and real data sets, we show that our proposed method performs competitively compared to a set of alternative state-of-the-art imbalanced classification algorithms.

Keywords: kernel Fisher discriminant analysis, imbalanced data, two-class classification

1. Introduction

Classification is a task of supervised learning in which the response function assumes a set of integer values known as the class labels. In particular, two-class classification refers to algorithms producing binary responses and aiming at separating two probability densities after observing some instances from each class. In this paper, we are interested in developing a classification algorithm for a two-class classification problem in which the number of data points in one class (i.e. the majority class) is greater than those of the other class (i.e. the minority class). This type of data structure is called imbalanced data.

It is particularly crucial to correctly identify test cases belonging to the minority class as a low detection rate for the minority class could incur heavy expenses in practice. The reason lies in the nature of the minority classes. For example, in quality control applications,

the minority class is the class of defective products; in security applications, the minority class is the class of potential perpetrators or attackers; in medical applications, the minority class is the class of diseases or cancerous cells. A classification method that fails to detect the minority classes is useless for practical purposes.

If one is interested in detecting minority cases in application, a direct use of traditional two-class classifications, such as support-vector machines or logistic regression, is not reliable because when the minority class data are too few in the training set, those methods tend to label almost all the instances in the test set, minority or otherwise, as the majority class (Chen et al., 2005). A training data set overwhelmed with one class of data points and deficient in the other class misleads the two classification algorithms about the accurate boundary between the two groups. Using most standard loss functions, these classification algorithms see little penalty by classifying regions in which both the minority and majority points have high density.

The major efforts aimed at solving the imbalanced classification problem can be categorized into: (a) cost-sensitive methods and (b) sampling strategies (He and Garcia, 2009; Japkowicz, 2000). Cost-sensitive methods take the imbalance structure into account by assigning a higher cost to the miss-classification of minority data points (Elkan, 2001; Ting, 2002). Despite a theoretical connection between imbalanced structure and cost-sensitive framework (Maloof, 2003; Weiss, 2004), this class of algorithms however may fail in practice; for example, if in the training stage the instances forming the classes are separable (Wallace et al., 2011, p.757). More critically, determining a suitable cost function is not a straightforward task and it may be difficult to achieve a robust algorithm using cost-sensitive methods.

The basic idea of the sampling-based approach is to alter the imbalanced structure of the problem by using different types of sampling methods. Hence, the algorithms in this category can be classified according to the specific sampling approaches, including resampling with replication, undersampling, or synthetic oversampling. In resampling with replication, one can use, for instance, bootstrapping for oversampling the minority data (Chen et al., 2005; Byon et al., 2010). In undersampling, one downsamples the majority data points to create more balanced data sets and alleviate the imbalance attached to the original data (Liu et al., 2009).

A novel approach proposed by Chawla et al. (2002) and called SMOTE, generates extra synthetic minority data points by interpolating the spaces between existing minority data points. Unlike other sampling methods which resample the existing data, SMOTE "creates" new data points, debuting the synthetic oversampling approach. Since SMOTE, many other variations of synthetic oversampling have been proposed in the literature; among others, Han et al. (2005) proposed an algorithm generating minority data points close to the boundary of the two classes and Batista et al. (2004) utilized different heuristics to integrate with synthetic oversampling.

SMOTE has proven to be a powerful method for handling imbalanced classification problems and still serves as a benchmark for this class of problems. An important revelation from the success of SMOTE and the like is that the synthetic oversampling is more potent than merely resampling existing data. The power of synthetic oversampling seems to lie in the simple fact that extra data are synthesized. From another perspective, synthetic data generation can be considered as a case of "phantom-transduction" as opposed to

the inductive inference (Akbani et al., 2004). In other words, generating extra synthetic minority data points resembles that of using test sets in learning (Gammerman et al., 1998). SMOTE, for instance, does not employ a sophisticated approach for data synthesizing, but uses a simple, yet proved highly effective in practice, data interpolation (Chawla et al., 2002). It is not clear, however, whether the mechanism of data synthesizing matters and if so, which type of mechanism to use.

The current literature does not seem to present a consensus concerning the effectiveness of data synthesizing mechanisms. We tend to believe that it matters, because if a data synthesizing mechanism is tailored to and/or embedded in a specific classification problem, we expect to observe improvements in classification performance. Some empirical evidence supports our belief (Han et al., 2005). At a minimum, we believe that the data synthesizing issue remains unsettled and is worth further investigation.

We also believe that an important question to ponder is how to decide the decision boundary if we were furnished with more instances of the minority class. It should be emphasized, however, that not all those could-be minority points carry the same amount of information; those that can guide the algorithm to expand the minority class's region are more valuable because it is the difficulty that classification algorithms confront. Basically the question becomes how to use the current data points to synthesize the "valuable" but absent minority data points that allow us to obtain a tighter boundary for the majority class.

Towards that goal, we employ the kernel trick embedded in Fisher discriminant analysis (Hofmann et al., 2008; Mika et al., 1999) in our data synthesizing mechanism in order to exploit the properties of newly generated points in the feature space without actually specifying them. We utilize two properties of the "artificially" generated minorities: (i) the points should be located as close as possible to the boundary of the majority class, (ii) their projection onto a lower dimensional space should be close to that of the existing minority points in their vicinity. Then we sample more minority points from the augmented data set, conditional on the boundary achieved. We perform this procedure iteratively until the algorithm achieves the desired performance, and label the resulting algorithm Absent Data Generator (ADG).

The remainder of this paper is organized as follows. Section 2 outlines the kernel Fisher discriminant analysis, formally defines the imbalanced classification problem and presents the main optimization formulation. Section 3 presents the details of the proposed algorithm. Section 4 describes the proposed method's application to several simulated and real data sets and the results when the data structure is imbalanced. Section 5 discusses finding a bound on the generalization error of the algorithm. Section 6 concludes the paper and offers suggestions for future research.

2. Problem Formulation

Let $\mathcal{X}$ denote the input space, and suppose $\mathcal{X}^- = \{\boldsymbol{x}_1^-, \boldsymbol{x}_2^-, \ldots, \boldsymbol{x}_{l_-}^-\} \subset \mathcal{X}$ is the training set of majority data points which are independent and identically distributed (i.i.d.) (negative points, labeled as -1 or simply "$-$") and $\mathcal{X}^+ = \{\boldsymbol{x}_1^+, \boldsymbol{x}_2^+, \ldots, \boldsymbol{x}_{l_+}^+\} \subset \mathcal{X}$ is the training set of minority data points, also i.i.d. (positive points, labeled as $+1$, or simply "$+$"). For notation simplicity, the subscript "$+$" on l_+ is dropped when the context is clear. In the

case of imbalance data, we have $l_+ \ll l_-$, or simply $l \ll l_-$. The goal in this section is to introduce a basic framework and general thoughts on how to generate and then utilize artificial data points. We propose generating artificial data points near the discriminative boundary of the two classes, and that they are generated within existing clusters with the probability of artificial data generated within a cluster inversely proportional to the size of that cluster.

First, we introduce the notion of "absent data": intuitively, absent data refer to the data points belonging to the minority class whose lack of presence has made the problem imbalanced, and we intend to re-generate them for the purpose of two-class classification. The concept of some data being absent is based on the thought that the existing data points may convey some information that allows us to identify some new data points belonging to the same class. Of course, acknowledging the existing of absent data does not imply that we know their numbers or exact locations in the space *a priori*. But in the context of imbalanced classification, this assumption paves the way for solving the problem through synthetic oversampling of minority data. Let $\mathcal{Z} = \{\boldsymbol{x}^+_{l+1}, \boldsymbol{x}^+_{l+2}, \ldots, \boldsymbol{x}^+_{l+k}\} \subset \mathcal{X}^+$ denote these absent data from the minority class; we assume the absent data are also an i.i.d. sample. We may denote each $\boldsymbol{x}^+_{l+j} \in \mathcal{Z}$ by $\boldsymbol{z}_j$ for $j = 1, 2, \ldots, k$.

We first review the Fisher discriminant analysis briefly. For a two-class classification, Fisher linear discriminant can be expressed simply through the following optimization problem:

$$\max_{\boldsymbol{w}} J(\boldsymbol{w}) = \frac{\boldsymbol{w}^T \boldsymbol{S}_B \boldsymbol{w}}{\boldsymbol{w}^T \boldsymbol{S}_W \boldsymbol{w}}, \tag{1}$$

where $\boldsymbol{S}_B$ and $\boldsymbol{S}_W$ are the between and within class scatter matrices, respectively:

$$\boldsymbol{S}_B = (\boldsymbol{m}_- - \boldsymbol{m}_+)(\boldsymbol{m}_- - \boldsymbol{m}_+)^T,$$

$$\boldsymbol{S}_W = \sum_{i \in \{-,+\}} \sum_{\boldsymbol{x} \in \mathcal{X}^i} (\boldsymbol{x} - \boldsymbol{m}_i)(\boldsymbol{x} - \boldsymbol{m}_i)^T, \tag{2}$$

and $\boldsymbol{m}_i = \frac{1}{l_i} \sum_{j=1}^{l_i} \boldsymbol{x}^i_j$, for $i \in \{-, +\}$, is the sample average of each class. Problem (1) can be interpreted as maximizing the ratio of the between-class variance to the pooled variance about the means. Under certain conditions, we can also interpret this formulation as an optimal Bayes classifier (Bickel and Levina, 2004). We will revisit this formulation in Section 5 when developing an error bound.

To deal with nonlinear cases, one can map the data into a high-dimensional feature space and perform the calculation in that space. However, if an appropriate kernel is chosen for the transformation of the data and the calculation only requires kernel evaluations, we do not have to perform any calculations in the high-dimensional feature space (Hofmann et al., 2008). This property, known as the kernel trick, can be applied to the Fisher discriminant analysis, resulting in the Kernel Fisher Discriminant (KFD) (Mika et al., 1999). Specifically, the KFD is the extension of the Fisher linear discriminant performed in the feature space which solves

$$\max_{\boldsymbol{w}} J(\boldsymbol{w}) = \frac{\boldsymbol{w}^T \boldsymbol{S}^{\phi}_B \boldsymbol{w}}{\boldsymbol{w}^T \boldsymbol{S}^{\phi}_W \boldsymbol{w}}, \tag{3}$$

where $\boldsymbol{S}_B^{\boldsymbol{\phi}}$ and $\boldsymbol{S}_W^{\boldsymbol{\phi}}$ are the between and within class scatter matrices, respectively, in the feature space:

$$\boldsymbol{S}_B^{\boldsymbol{\phi}} = (\boldsymbol{m}_-^{\boldsymbol{\phi}} - \boldsymbol{m}_+^{\boldsymbol{\phi}})(\boldsymbol{m}_-^{\boldsymbol{\phi}} - \boldsymbol{m}_+^{\boldsymbol{\phi}})^T,$$
$$\boldsymbol{S}_W^{\boldsymbol{\phi}} = \sum_{i\in\{-,+\}} \sum_{\boldsymbol{x}\in\mathcal{X}^i} (\boldsymbol{\phi}(\boldsymbol{x}) - \boldsymbol{m}_i^{\boldsymbol{\phi}})(\boldsymbol{\phi}(\boldsymbol{x}) - \boldsymbol{m}_i^{\boldsymbol{\phi}})^T, \tag{4}$$

and $\boldsymbol{m}_i^{\boldsymbol{\phi}} = \frac{1}{l_i}\sum_{j=1}^{l_i} \boldsymbol{\phi}(\boldsymbol{x}_j^i)$. Here, $\boldsymbol{\phi}$ is a nonlinear mapping from $\mathcal{X}$ to the feature space $\mathcal{F}$, which is assumed to be a separable Hilbert space endowed with an inner product $\langle\cdot,\cdot\rangle$ such that there exists a function $K : \mathcal{X}\times\mathcal{X}\to\mathbb{R}$ where $K(\boldsymbol{x},\boldsymbol{x}') = \langle\boldsymbol{\phi}(\boldsymbol{x}),\boldsymbol{\phi}(\boldsymbol{x}')\rangle$. Obviously, in this case $\boldsymbol{w}\in\mathcal{F}$. Applying to imbalanced data sets, KFD suffers the same problem as most other classifiers do, i.e. it falls short of detecting most minority points in the test stage.

Our goal is to consider the imbalanced structure explicitly and extend KFD in such a way that it could be applied to imbalanced data. Towards this end, our thought process is as follows: first, if we had extra data points from the minority class, those points would be projected with high probability to where the existing minority points are projected; second, points close to the boundary of the majority points carry more "information" so we can use them to find the separating hyperplane in the feature space. The latter is in fact an intuitive property we are seeking, but the former requires more clarification. Particularly, if dealing with complex patterns in high dimensions, we may frequently observe that the minority data points constitute separate clusters after (or before) being projected to a lower-dimensional space. Therefore, if resemblance in projection regions is used as a property to generate artificial data, it entails precaution against the effect of complex structures. One way to address this issue is to take the cluster-based structure of the data into account explicitly.

Suppose the training minority points constitute C different clusters, for $C\geq 1$. That is, we have $\mathcal{X}^+ = \bigcup_{c=1}^C \mathcal{X}_c^+$ and $\mathcal{X}_{c'}^+\cap\mathcal{X}_c^+ = \emptyset$ for $c\neq c'$, where $\mathcal{X}_c^+ = \{\boldsymbol{x}_{1,c}^+, \boldsymbol{x}_{2,c}^+, \cdots, \boldsymbol{x}_{l_c,c}^+\}$ is the c-th cluster of the minority data points, and we have $|\mathcal{X}_c^+| = l_c$, and $\sum_{c=1}^C l_c = l$. Accordingly, we partition the absent data in $\mathcal{Z}$ also into C different clusters, $\mathcal{Z}_c$'s, for $c = 1,2,\ldots,C$, each of which corresponds to one of the C clusters of the minority points. Specifically $\mathcal{Z}_c = \{\boldsymbol{x}_{l_c+1,c}^+, \boldsymbol{x}_{l_c+2,c}^+, \ldots, \boldsymbol{x}_{l_c+k_c,c}^+\}$, $\bigcup_{c=1}^C \mathcal{Z}_c = \mathcal{Z}$, and $\mathcal{Z}_{c'}\cap\mathcal{Z}_c = \emptyset$, for $c\neq c'$. We also have $|\mathcal{Z}_c| = k_c$, and $\sum_{c=1}^C k_c = k$. The previously defined notation $\boldsymbol{z}_j$ can be similarly extended as $\boldsymbol{z}_{j,c} := \boldsymbol{x}_{l_c+j,c}^+$, for $j = 1,2,\cdots,k_c$.

To enforce the property that newly generated points would be projected with high probability to where the existing minority points are projected, we add the constraint

$$\left(\boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{z}_{j,c}) - \boldsymbol{w}^T\boldsymbol{m}_{+,c}^{\boldsymbol{\phi}}\right)^2 \leq \delta, \quad \text{for} \quad j = 1,2,\ldots k_c, \quad c = 1,2,\ldots C, \tag{5}$$

for some positive $\delta > 0$, where $\boldsymbol{m}_{+,c}^{\boldsymbol{\phi}} = \frac{1}{l_c}\sum_{j=1}^{l_c}\boldsymbol{\phi}(\boldsymbol{x}_{j,c}^+)$, namely the mean of cluster c in the feature space. To have the second property, i.e. to have more points close to the boundary of the majority points, we add another constraint,

$$(\boldsymbol{\phi}(\boldsymbol{z}_{j,c}) - \boldsymbol{m}_-^{\boldsymbol{\phi}})^T(\boldsymbol{\phi}(\boldsymbol{z}_{j,c}) - \boldsymbol{m}_-^{\boldsymbol{\phi}}) \leq \Lambda \quad \text{for} \quad j = 1,2,\ldots k_c, \quad c = 1,2,\ldots C, \tag{6}$$

for some positive $\Lambda > 0$. Constraint (5) ensures that the point $\boldsymbol{\phi}(\boldsymbol{z}_{j,c})$ is at most δ distance away from the current cluster center of a minority group. This constraint also incorporates the cases where the minority data are cohesive and do not constitute many clusters, i.e.

only one cluster is determined according to the algorithm discussed in Section 3, which means that the constraint implies that the newly generated data point is at most δ distance away from the mean of the minority data points. Constraint (6) ensures that the newly generated points are "useful" in the sense that they are located close to the boundary of the two groups.

As a result of the Representer's Theorem (Hofmann et al., 2008), we can safely assume both $\boldsymbol{w}$ and $\boldsymbol{\phi}(\boldsymbol{z}_{j,c})$ belong to the space generated by the training points, namely $\mathcal{X}^- \cup \mathcal{X}^+$, whose elements, with a slight abuse of notation, can be represented by $\{\boldsymbol{x}_p\}_{p=1}^n$ where $n = l_- + l$. Specifically,

$$\boldsymbol{w} = \sum_{p=1}^{n} \alpha_p \boldsymbol{\phi}(\boldsymbol{x}_p), \tag{7}$$

and

$$\boldsymbol{\phi}(\boldsymbol{z}_{j,c}) - \boldsymbol{m}_-^{\boldsymbol{\phi}} = \sum_{p=1}^{n} \beta_p^{j,c} \boldsymbol{\phi}(\boldsymbol{x}_p), \quad \text{for} \quad j = 1, 2, \ldots k_c, \quad c = 1, 2, \ldots C, \tag{8}$$

where α_p and $\beta_p^{j,c}$ are real coefficients for $p = 1, 2, \ldots, n$, $j = 1, 2, \ldots, k_c$ and $c = 1, 2, \ldots C$. Having made these assumptions, we can express constraints (5) and (6) as

$$\left[\sum_{p=1}^{n} \alpha_p \boldsymbol{\phi}(\boldsymbol{x}_p)^T \left(\sum_{p=1}^{n} \beta_p^{j,c} \boldsymbol{\phi}(\boldsymbol{x}_p) + \frac{1}{l_-} \sum_{\ell=1}^{l_-} \boldsymbol{\phi}(\boldsymbol{x}_\ell^-) - \frac{1}{l_c} \sum_{\ell=1}^{l_c} \boldsymbol{\phi}(\boldsymbol{x}_\ell^+) \right) \right]^2 \leq \delta, \tag{9}$$

and

$$\sum_{p=1}^{n} (\beta_p^{j,c})^2 K(\boldsymbol{x}_p, \boldsymbol{x}_p) \leq \Lambda, \quad \text{for} \quad j = 1, 2, \ldots k_c, \quad c = 1, 2, \ldots C, \tag{10}$$

respectively. In the matrix forms, the above two expressions can be represented as

$$\left[\boldsymbol{\alpha}^T \boldsymbol{K} \boldsymbol{\beta}^{j,c} + \boldsymbol{\alpha}^T (\boldsymbol{M}_- - \boldsymbol{M}_+^c) \right]^2 \leq \delta, \quad \text{for} \quad j = 1, 2, \ldots k_c, \quad c = 1, 2, \ldots C, \tag{11}$$

and

$$(\boldsymbol{\beta}^{j,c})^T \boldsymbol{K} (\boldsymbol{\beta}^{j,c}) \leq \Lambda, \quad \text{for} \quad j = 1, 2, \ldots k_c, \quad c = 1, 2, \ldots C, \tag{12}$$

where $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_n]^T$ and $\boldsymbol{\beta}^{j,c} = [\beta_1^{j,c}, \beta_2^{j,c}, \ldots \beta_n^{j,c}]^T$, and $\boldsymbol{M}_-$ is an $n \times 1$ vector such that $(\boldsymbol{M}_-)_j = \frac{1}{l_-} \sum_{\ell=1}^{l_-} K(\boldsymbol{x}_j, \boldsymbol{x}_\ell^-)$, and $\boldsymbol{M}_+^c$ is an $n \times 1$ vector such that $(\boldsymbol{M}_+^c)_j = \frac{1}{l_c} \sum_{\ell=1}^{l_c} K(\boldsymbol{x}_j, \boldsymbol{x}_{\ell,c}^+)$. The $n \times n$ matrix $\boldsymbol{K}$ consists of all of the pairwise kernel evaluations, namely $(\boldsymbol{K})_{r,s} = K(\boldsymbol{x}_r, \boldsymbol{x}_s)$, for $r, s \in \{1, 2, \ldots, n\}$.

Following the notation introduced in Mika et al. (1999),

$$\boldsymbol{M} := (\boldsymbol{M}_- - \boldsymbol{M}_+)(\boldsymbol{M}_- - \boldsymbol{M}_+)^T, \text{ and} \tag{13}$$

$$\boldsymbol{N} := \sum_{i \in \{-,+\}} \boldsymbol{K}_i (\boldsymbol{I} - \boldsymbol{1}_{l_i}) \boldsymbol{K}_i^T, \tag{14}$$

where $\boldsymbol{M}_+$ is an $n \times 1$ vector such that $(\boldsymbol{M}_+)_j = \frac{1}{l} \sum_{\ell=1}^{l} K(\boldsymbol{x}_j, \boldsymbol{x}_\ell^+)$, $\boldsymbol{K}_i$ is an $n \times l_i$ matrix with $(\boldsymbol{K}_i)_{r,s} = K(\boldsymbol{x}_r, \boldsymbol{x}_s^i)$ for $r \in \{1, 2, \ldots, n\}$, $s \in \{1, 2, \ldots, l_i\}$ for $i \in \{-, +\}$, $\boldsymbol{I}$ is the

identity matrix of appropriate size, and $\mathbf{1}_{l_i}$ is a matrix of appropriate size whose entries are $\frac{1}{l_i}$ for $i = -$ and $i = +$, respectively. Now, we can formulate the classification problem with imbalanced data through the following optimization

$$\max_{\boldsymbol{\alpha}} J(\boldsymbol{\alpha}) = \frac{\boldsymbol{\alpha}^T \boldsymbol{M} \boldsymbol{\alpha}}{\boldsymbol{\alpha}^T \boldsymbol{N} \boldsymbol{\alpha}}, \tag{15}$$

subject to

$$\left[\boldsymbol{\alpha}^T \boldsymbol{K} \boldsymbol{\beta}^{j,c} + \boldsymbol{\alpha}^T (\boldsymbol{M}_- - \boldsymbol{M}_+^c)\right]^2 \leq \delta, \quad \text{for} \quad j = 1, 2, \dots k_c, \quad c = 1, 2, \dots C, \tag{16}$$

$$(\boldsymbol{\beta}^{j,c})^T \boldsymbol{K} (\boldsymbol{\beta}^{j,c}) \leq \Lambda, \quad \text{for} \quad j = 1, 2, \dots k_c, \quad c = 1, 2, \dots C. \tag{17}$$

To solve optimization problem (15)-(17), we assume $\delta = 0$. This implies that the newly generated points $\boldsymbol{z}_{j,c}$ should be projected where the mean of the corresponding cluster in the minority group is projected. As such, constraint (16) is replaced by

$$\boldsymbol{\alpha}^T \boldsymbol{K} \boldsymbol{\beta}^{j,c} + \boldsymbol{\alpha}^T (\boldsymbol{M}_- - \boldsymbol{M}_+) = 0, \quad \text{for} \quad j = 1, 2, \dots k_c, \quad c = 1, 2, \dots C.$$

This new constraint is not restricting, since we next solve a relaxation of the original problem. Specifically, we use the Lagrangian relaxation (Anstreicher and Wolkowicz, 1998) for solving ((15))-(17). First, note that an equivalent way of writing the optimization ((15))-(17) is to consider the denominator in the objective function (15) as another constraint and only have the numerator in the objective function. Specifically, we consider the objective function to be

$$\max_{\boldsymbol{\alpha}} J(\boldsymbol{\alpha}) = \boldsymbol{\alpha}^T \boldsymbol{M} \boldsymbol{\alpha}, \tag{18}$$

and add the constraint

$$\boldsymbol{\alpha}^T \boldsymbol{N} \boldsymbol{\alpha} \leq R, \tag{19}$$

to the optimization problem (15), for some positive number R. Having done that, we get the following for the Lagrangian function

$$\begin{aligned} J(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \boldsymbol{\alpha}^T \boldsymbol{M} \boldsymbol{\alpha} \quad &- \quad \gamma \left[\boldsymbol{\alpha}^T \boldsymbol{N} \boldsymbol{\alpha} - R\right] \\ &- \quad \sum_{c=1}^{C} \sum_{j=1}^{k_c} \lambda_j^c \left[\boldsymbol{\alpha}^T \boldsymbol{K} \boldsymbol{\beta}^{j,c} + \boldsymbol{\alpha}^T (\boldsymbol{M}_- - \boldsymbol{M}_+^c)\right] \\ &- \quad \sum_{c=1}^{C} \sum_{j=1}^{k_c} \mu_j^c \left[(\boldsymbol{\beta}^{j,c})^T \boldsymbol{K} (\boldsymbol{\beta}^{j,c}) - \Lambda\right], \end{aligned} \tag{20}$$

for $\gamma, \lambda_j^c, \mu_j^c > 0$.

To find the stationary points, we set the partial derivatives of the Lagrangian to zero,

$$\frac{\partial}{\partial \boldsymbol{\alpha}} J = 2 \left(\boldsymbol{M} - \gamma \boldsymbol{N}\right) \boldsymbol{\alpha} - \sum_{c=1}^{C} \sum_{j=1}^{k_c} \lambda_j^c \left(\boldsymbol{K} \boldsymbol{\beta}^{j,c} + (\boldsymbol{M}_- - \boldsymbol{M}_+^c)\right) = 0, \tag{21}$$

$$\frac{\partial}{\partial \boldsymbol{\beta}^{j,c}} J = -\lambda_j^c \left(\boldsymbol{K}\boldsymbol{\alpha}\right) - 2\mu_j^c \boldsymbol{K}\boldsymbol{\beta}^{j,c} = 0, \quad \text{for} \quad j = 1, 2, \ldots k_c, \quad c = 1, 2, \ldots C. \tag{22}$$

Substituting $\boldsymbol{\beta}^{j,c} = -\frac{\lambda_j^c}{2\mu_j^c}\boldsymbol{\alpha}$, which results from (22), into (21) yields

$$2\left(\boldsymbol{M} - \gamma \boldsymbol{N}\right)\boldsymbol{\alpha} = -\sum_{c=1}^{C}\sum_{j=1}^{k_c} \lambda_j^c \left(\boldsymbol{K}\frac{\lambda_j^c}{2\mu_j^c}\boldsymbol{\alpha} + (\boldsymbol{M}_- - \boldsymbol{M}_+^c)\right), \tag{23}$$

which can be further simplified as

$$\left(\boldsymbol{M} - \gamma \boldsymbol{N}\right)\boldsymbol{\alpha} = -\boldsymbol{K}\boldsymbol{\alpha}\sum_{c=1}^{C}\sum_{j=1}^{k_c}\frac{(\lambda_j^c)^2}{4\mu_j^c} - \sum_{c=1}^{C}\left\{(\boldsymbol{M}_- - \boldsymbol{M}_+^c)\sum_{j=1}^{k_c}\frac{\lambda_j^c}{2}\right\}. \tag{24}$$

Let $\omega = -\sum_{c=1}^{C}\sum_{j=1}^{k_c}\frac{(\lambda_j^c)^2}{4\mu_j^c}$, and $\nu^c = -\sum_{j=1}^{k_c}\frac{\lambda_j^c}{2}$. Then, we have

$$\left(\boldsymbol{M} - \gamma \boldsymbol{N}\right)\boldsymbol{\alpha} = \boldsymbol{K}\boldsymbol{\alpha}\omega + \sum_{c=1}^{C}\left\{(\boldsymbol{M}_- - \boldsymbol{M}_+^c)\nu^c\right\}. \tag{25}$$

Therefore, $\boldsymbol{\alpha}$ is the solution of the linear system

$$\left((\boldsymbol{M} - \gamma \boldsymbol{N}) - \omega\boldsymbol{K}\right)\boldsymbol{\alpha} = \sum_{c=1}^{C}\left\{(\boldsymbol{M}_- - \boldsymbol{M}_+^c)\nu^c\right\}. \tag{26}$$

Solving the problem yields the optimal projection coefficients $\boldsymbol{\alpha}^* = [\alpha_1^*, \alpha_2^*, \ldots, \alpha_n^*]$. Subsequently, we find the projection of a new test point $\boldsymbol{x}_{\text{test}} \subset \mathcal{X}$ onto $\boldsymbol{w}$ by

$$< \boldsymbol{w}, \boldsymbol{\phi}(\boldsymbol{x}_{\text{test}}) > = \sum_{\ell=1}^{l} \alpha_\ell^* K(\boldsymbol{x}_\ell, \boldsymbol{x}_{\text{test}}). \tag{27}$$

The solution of the linear system of equations, namely (26), provides us with the coefficients $\boldsymbol{\alpha}^*$ which will be used for finding a tighter boundary for the majority class. We note that despite not being present in (26), $\boldsymbol{\beta}^{j,c}$'s affect the values of $\boldsymbol{\alpha}$ through the values of the Lagrangian coefficients, λ_j^c and μ_j^c. As such, $\boldsymbol{\beta}^{j,c}$'s are used implicitly to identify the locations of absent points, although not explicitly needed for prediction; this is how the use of absent points helps find a tighter boundary for the majority class.

3. Algorithm

As mentioned in Section 2, in optimization problem (15)-(17) we find the projection coefficients $\boldsymbol{\alpha}^*$ based upon two considerations specifically developed to address the imbalanced structure. The outcome is a decision boundary separating the two classes. Yet, those absent data points are still implicitly considered and have not been used to update the estimate of scatter matrices. This section explains how to generate the synthetic data points, based on the newly decided class boundary, and use them to update the scatter matrices.

From a different angle, optimization problem (15)-(17) can be seen as a way to expand the region associated with the minority data points, as opposed to the region one would have had without imposing constraints (16) and (17). This expansion allows us to identify the minority region with better precision and to estimate $\boldsymbol{S}_B$ and $\boldsymbol{S}_W$ more accurately. Once the minority class region is revised, we can use the knowledge to synthesize more data points for the minority class.

We start by using an iterative procedure that alternately updates the class boundary and revises the $\boldsymbol{S}_B$ and $\boldsymbol{S}_W$ estimation. In other words, optimization problem (15)-(17) splits the input region $\mathcal{X}$ into two disjoint regions $\mathcal{X}^-$ and $\mathcal{X}^+$ which are estimated regions belonging to the majority and minority points, respectively. Then, we draw additional minority points, i.e. data synthesizing for the minority class, from the updated minority region to improve the estimates of the scatter matrices.

Specifically, we draw independent samples from the estimated density of the current minority points, conditional on the boundary imposed by the optimal projection coefficients $\boldsymbol{\alpha}_*$. Let $\widehat{F}^u_{\boldsymbol{\alpha}_*}$ be the estimated distribution of the minority points as a mixture of u Gaussian distribution estimated using $\mathcal{X}^+ = \{\boldsymbol{x}_1^+, \boldsymbol{x}_2^+, \ldots, \boldsymbol{x}_l^+\} \subset \mathcal{X}$ and truncated according to $\boldsymbol{\alpha}_*$, namely

$$\widehat{F}^u_{\boldsymbol{\alpha}_*} = \frac{1}{u}\sum_{b=1}^{u} a_b \Psi_b, \tag{28}$$

where Ψ_b is a Gaussian distribution with mean $\boldsymbol{\mu}_b$ and variance $\boldsymbol{\Sigma}_b^2$, truncated over the region $\mathcal{X}^+$, $0 \le a_b \le 1$ for $b = 1, 2, \ldots, u$, and $\sum_{b=1}^{u} a_b = 1$. Let $\widetilde{\mathcal{Z}}$ denote a set of q independent samples drawn from $\widehat{F}^u_{\boldsymbol{\alpha}_*}$, specifically $\widetilde{\boldsymbol{x}}_\ell^+ \sim \widehat{F}^u_{\boldsymbol{\alpha}_*}$, for $\ell = 1, 2, \ldots, q$. Denote the augmented minority set by $\widetilde{\mathcal{X}}^+ = \mathcal{X}^+ \bigcup \widetilde{\mathcal{Z}} = \{\boldsymbol{x}_1^+, \boldsymbol{x}_2^+, \ldots, \boldsymbol{x}_l^+, \widetilde{\boldsymbol{x}}_1^+, \widetilde{\boldsymbol{x}}_2^+, \ldots, \widetilde{\boldsymbol{x}}_q^+\}$. Note the difference between $\widetilde{\boldsymbol{x}}_\ell^+$ used here and $\boldsymbol{z}_\ell$ used in the previous section: $\boldsymbol{z}_\ell$ denotes the absent data points close to the class boundary, playing a role similar to the support vector points, while $\widetilde{\boldsymbol{x}}_\ell^+$ denotes any data point actually generated for the minority class. The $\widetilde{\boldsymbol{x}}_\ell^+$ points cannot be guaranteed to be close to the class boundary; rather they may be over the interior of the minority region or cross the boundary and over the region of the majority class (called intrusion). Consequently, there is a difference between k and q: k is the number of data points represented by $\boldsymbol{z}_\ell$, similar to the number of support vector points, while q is the number of actually generated data points scattering around in the input space. Generally, q is larger than k.

Then we use the augmented minority set to reevaluate the between- and within-class scatter matrices, such as:

$$\widetilde{\boldsymbol{S}}_B^{\boldsymbol{\phi}} = \left(\boldsymbol{m}_-^{\boldsymbol{\phi}} - \widetilde{\boldsymbol{m}}_+^{\boldsymbol{\phi}}\right)\left(\boldsymbol{m}_-^{\boldsymbol{\phi}} - \widetilde{\boldsymbol{m}}_+^{\boldsymbol{\phi}}\right)^T,$$

$$\widetilde{\boldsymbol{S}}_W^{\boldsymbol{\phi}} = \sum_{\boldsymbol{x}\in\mathcal{X}^-} (\boldsymbol{\phi}(\boldsymbol{x}) - \boldsymbol{m}_-^{\boldsymbol{\phi}})(\boldsymbol{\phi}(\boldsymbol{x}) - \boldsymbol{m}_-^{\boldsymbol{\phi}})^T + \sum_{\boldsymbol{x}\in\widetilde{\mathcal{X}}^+} \left(\boldsymbol{\phi}(\boldsymbol{x}) - \widetilde{\boldsymbol{m}}_+^{\boldsymbol{\phi}}\right)\left(\boldsymbol{\phi}(\boldsymbol{x}) - \widetilde{\boldsymbol{m}}_+^{\boldsymbol{\phi}}\right)^T, \tag{29}$$

where $\widetilde{\boldsymbol{m}}_+^{\boldsymbol{\phi}} = \frac{1}{l+q}\sum_{j=1}^{l+q} \boldsymbol{\phi}(\boldsymbol{x}_j^+)$. In other words, we update the estimates of the scatter matrices using the newly generated points. Using (7) and (8) and following the steps for the optimization procedure stated in Section 2, we obtain a new optimization problem similar to (15)-(17) in which the matrices $\boldsymbol{K}$, $\boldsymbol{N}$, and $\boldsymbol{M}$ and vectors $\boldsymbol{M}_-$ and $\boldsymbol{M}_+^c$, for

$c = 1, 2, \ldots, C$, are evaluated using the sets $\mathcal{X}^-$ and $\widetilde{\mathcal{X}}^+$. The new optimization problem yields a new optimal projection coefficient vector $\boldsymbol{\alpha}_*$ which, in turn, we use to re-estimate the scatter matrices by fitting again a mixture of Gaussian distributions and generating $q \leftarrow \lfloor \frac{q}{2} \rfloor$ absent points (i.e. half of the points we generated in the previous iteration). We continue this procedure until $q < 1$, and we use the final $\boldsymbol{\alpha}_*$ as the optimal projection coefficient vector.

The clusters at each stage are decided based on the X-means algorithm (Pelleg and Moore, 2000). X-means is simply a k-means clustering algorithm in which the number of clusters, which is denoted by C in our algorithm, is decided based on a Bayesian Information Criterion (BIC) (Hastie et al., 2009). We choose X-means because the number of clusters is not known in advance; this number is estimated by X-means based on data; other clustering methods can also be used (Fraley and Raftery, 1998).

The number of Gaussian mixtures to estimate the distribution of the minority points is also decided based on BIC. Specifically, the number of Gaussian mixtures at each iteration is

$$\arg\min_{u \in \mathbb{N}} \text{BIC}\left(\widehat{F}^u_{\boldsymbol{\alpha}_*}\right), \tag{30}$$

where $\mathbb{N}$ is the set of positive integers and

$$\text{BIC}\left(\widehat{F}^u_{\boldsymbol{\alpha}_*}\right) = -2\log(L) + u\log(q),$$

where L is the likelihood of the minority data points, assuming they are random samples from $\widehat{F}^u_{\boldsymbol{\alpha}_*}$.

Once we find the number of Gaussian mixtures, we generate q data points such that those points are sampled from the fitted Gaussian mixture, assuming the current boundary defined by the classifier. Among the q synthetic data points at each stage, we first admit $q' \leq q$ of them based on (27); this step is to discard the synthetic data points that are on the wrong side of the decision boundary. We denote the set of the admitted points by $\widetilde{\mathcal{Z}'}$, which is the final set of the newly generated data points at a given stage.

The data points in $\widetilde{\mathcal{Z}'}$ are then assigned to a cluster $c = 1, 2, \ldots, C$ according to their Euclidean distance to the center of the cluster in the original space. Specifically, for $\widetilde{\boldsymbol{x}}^+_\ell \in \widetilde{\mathcal{Z}'}$, its cluster membership is assigned as

$$c = \arg\min_{c' \in \{1,\ldots,C\}} \|\widetilde{\boldsymbol{x}}^+_\ell - \bar{\boldsymbol{x}}^+_{c'}\|, \tag{31}$$

where $\bar{\boldsymbol{x}}^+_{c'} = \frac{1}{l_{c'}} \sum_{\ell=1}^{l_{c'}} \boldsymbol{x}^+_{\ell,c'}$, which is the center of cluster c' in the original space. This gives us q_c new data points for each cluster $c = 1, 2, \ldots, C$, such that $\sum_{c=1}^{C} q_c = q'$.

The values of Lagrangian coefficients λ^j_c and μ^j_c are determined to be inversely proportional to the number of current minority data points in their associated clusters, namely $\lambda^j_c = \frac{\lambda}{l_c}$ and $\mu^j_c = \frac{\mu}{l_c}$. This means that if there are very few data points in a cluster, violating constraints (16) and (17) is more heavily penalized in comparison to the case when there are more data points in that cluster. The number of perceived absent data points in a cluster k_c is also inversely proportional to the current number of data points in that cluster, because a cluster formed by very few data points is not reliable enough to generate many new data points. Note that k_c is the *a priori* number of perceived absent data points in a cluster, while q_c is the actually generated data points belonging to cluster c.

Assuming we know the values of the tuning parameters γ and λ, we can summarize the steps of the Absent Data Generator classifier (ADG) in Algorithm 1. In practice, the aforementioned tuning parameters are determined using cross validation (Hastie et al., 2009). Based on our experiments, ADG is not very sensitive to the number of absent points k, so that it can be simply set to a number between 10 to 15. The number of actual minority data points generated, q, on the other hand, is decided so that the final data set of interest is relatively balanced. Note that the number of newly generated points q is decreasing at each stage.

Algorithm 1 Absent Data Generator for Imbalanced Classification

Given $\mathcal{X}^-$ and $\mathcal{X}^+$, evaluate $\boldsymbol{K}$, $\boldsymbol{M}$, $\boldsymbol{N}$, $\boldsymbol{K}_i$, and $\boldsymbol{M}_i$, for $i \in \{-,+\}$ and let $\widetilde{\mathcal{X}}^+ = \mathcal{X}^+$.

repeat

1. Find C clusters for the augmented minority set $\widetilde{\mathcal{X}}^+$, where C is decided by minimizing the associated BIC.
2. Choose $\lambda_c^j = \frac{\lambda}{l_c}$, $\mu_c^j = \frac{\mu}{l_c}$, for $j = 1, 2, \ldots k_c$, and k_c is chosen proportionally to $\frac{1}{l_c}$, for $c = 1, 2, \ldots C$, such that $\sum_{c=1}^{C} k_c = k$.
3. Let $\omega = -\sum_{c=1}^{C}\sum_{j=1}^{k_c} \frac{(\lambda_j^c)^2}{4\mu_j^c}$, $\nu^c = -\sum_{j=1}^{k_c} \frac{\lambda_j^c}{2}$ and $(\boldsymbol{M}_+^c)_j = \frac{1}{l_c}\sum_{\ell=1}^{l_c} K(\boldsymbol{x}_j, \boldsymbol{x}_{\ell,c}^+)$
4. Let $\boldsymbol{\alpha}_*$ be the solution of $((\boldsymbol{M} - \gamma\boldsymbol{N}) - \omega\boldsymbol{K})\,\boldsymbol{\alpha} = \sum_{c=1}^{C}\left\{(\boldsymbol{M}_- - \boldsymbol{M}_+^c)\nu^c\right\}$.
5. Fit a mixture of u normal distributions to $\mathcal{X}^+$ where u provides the smallest BIC in (30).
6. Generate q data points from the resulting Gaussian mixtures above, say $\widetilde{\mathcal{Z}} = \{\widetilde{\boldsymbol{x}}_1^+, \widetilde{\boldsymbol{x}}_2^+, \ldots, \widetilde{\boldsymbol{x}}_q^+\}$.
7. Utilize $\boldsymbol{\alpha}_*$ according to (27) to test if each $\widetilde{\boldsymbol{x}}_\ell^+$, $\ell = 1, 2, \ldots, q$ belongs to class $+1$ or not. Let $\widetilde{\mathcal{Z}'} = \{x_{l+1}^+, \boldsymbol{x}_{l+2}^+, \ldots, \boldsymbol{x}_{l+q'}^+\} \subset \widetilde{\mathcal{Z}}$ be the set of data points admitted into the minority set.
8. Identify the clusters to which the new data points belong according to (31). Let q_c be the number of elements in $\widetilde{\mathcal{Z}'}$ belonging to cluster c, for $c = 1, 2, \ldots C$.
9. $\widetilde{\mathcal{X}}^+ \leftarrow \widetilde{\mathcal{X}}^+ \cup \widetilde{\mathcal{Z}'}$.
10. $\widetilde{\mathcal{X}} \leftarrow \mathcal{X}^- \cup \widetilde{\mathcal{X}}^+$.
11. $(\boldsymbol{M}_-)_j \leftarrow \frac{1}{l_-}\sum_{\ell=1}^{l_-} K(\boldsymbol{x}_j, \boldsymbol{x}_\ell^-)$ for $\boldsymbol{x}_j \in \widetilde{\mathcal{X}}$.
12. $(\boldsymbol{M}_+)_j \leftarrow \frac{1}{l+q'}\sum_{\ell=1}^{l+q'} K(\boldsymbol{x}_j, \boldsymbol{x}_\ell^+)$ for $\boldsymbol{x}_j \in \widetilde{\mathcal{X}}$.
13. $(\boldsymbol{M}_+^c)_j = \frac{1}{l_c+q_c}\sum_{\ell=1}^{l_c+q_c} K(\boldsymbol{x}_j, \boldsymbol{x}_{\ell,c}^+)$ for $\boldsymbol{x}_j \in \widetilde{\mathcal{X}}$.
14. $(\boldsymbol{K})_{r,s} \leftarrow K(\boldsymbol{x}_r, \boldsymbol{x}_s)$, for $r, s \in \{1, 2, \ldots, n+q'\}$,
    $(\boldsymbol{K}_-)_{r,s} = K(\boldsymbol{x}_r, \boldsymbol{x}_s^-)$ for $r \in \{1, 2, \ldots, n+q\}$, $s \in \{1, 2, \ldots, l_-\}$,
    $(\boldsymbol{K}_+)_{r,s} = K(\boldsymbol{x}_r, \boldsymbol{x}_s^+)$ for $r \in \{1, 2, \ldots, n+q\}$, $s \in \{1, 2, \ldots, l+q\}$.
15. $\boldsymbol{M} \leftarrow (\boldsymbol{M}_- - \boldsymbol{M}_+)(\boldsymbol{M}_- - \boldsymbol{M}_+)$.
16. $\boldsymbol{N} \leftarrow \sum_{i\in\{-,+\}} \boldsymbol{K}_i(\boldsymbol{I} - \boldsymbol{1}_{l_i})\boldsymbol{K}_i^T$.
17. $q \leftarrow \lfloor \frac{q}{2} \rfloor$,
    $l \leftarrow |\widetilde{\mathcal{X}}^+|$,
    $n \leftarrow |\widetilde{\mathcal{X}}|$.

until $q < 1$.

Having the optimal projection coefficients $\boldsymbol{\alpha}_*$ which corresponds to the optimal projection vector $\boldsymbol{w}$ in the feature space, we can obtain the prediction for the class labels by classifying the projected values of the data points onto $\boldsymbol{w}$. Let $\boldsymbol{\kappa}_{\boldsymbol{x}}$ be an $n \times 1$ vector of the kernel evaluation between $\boldsymbol{x} \in \mathcal{X}$ and all the training samples and the synthetic minority points generated by Algorithm 1, that is,

$$(\boldsymbol{\kappa}_{\boldsymbol{x}})_\ell = K(\boldsymbol{x}_\ell, \boldsymbol{x}), \quad \forall \boldsymbol{x}_\ell \in \widetilde{\mathcal{X}}. \tag{32}$$

Then assume $C_{\mathcal{T}}$ is a one-dimensional binary classifier, e.g. the Support Vector Machine, trained on the set $\mathcal{T} = \{(h(\boldsymbol{x}_\ell; \boldsymbol{\alpha}_*), y_\ell) : \boldsymbol{x}_\ell \in \widetilde{\mathcal{X}}, \ell = 1, 2, \ldots, n\}$, where y_ℓ is the class label for $\boldsymbol{x}_\ell$, and $h(\boldsymbol{x}_\ell; \boldsymbol{\alpha}_*) = \boldsymbol{\alpha}_*^T \boldsymbol{\kappa}_{\boldsymbol{x}_\ell}$. More precisely, the classifier $C_{\mathcal{T}}$, after training on $\mathcal{T}$, yields a real number as the threshold v_* such that if $h(\boldsymbol{x}; \boldsymbol{\alpha}_*) > v_*$, then the corresponding $h(\boldsymbol{x}; \boldsymbol{\alpha}_*)$ is labeled as $+1$; otherwise -1. Then, the label prediction for a test point $\boldsymbol{x}_t$ using the ADG will be

$$\text{ADG}(\boldsymbol{x}_t) = \begin{cases} +1 & \text{if } h(\boldsymbol{x}_t; \boldsymbol{\alpha}_*) > v_*, \\ -1 & \text{if } h(\boldsymbol{x}_t; \boldsymbol{\alpha}_*) \leq v_*. \end{cases} \tag{33}$$

We note that the ADG's data generation mechanism is based on an iterative method that explores the minority region by data generating constraints that are embedded in the optimization problem. Unlike SMOTE, in ADG, the synthetic data are not necessarily in the convex hull of existing data which could be another advantage for ADG, especially in higher dimensions. Also, ADG acknowledges the significance of the data points close to the boundary and generates synthetic data by utilizing both majority and minority data points.

ADG's computational complexity is of polynomial order. Note that the major operation in the algorithm is solving the system of linear equations (26), i.e. step 4 in the algorithm, since all the other steps involve relatively low computational costs. Particularly, clustering, if solved exactly, has cost $\mathcal{O}(n^{dC+1} \log n)$, where d denotes the dimension here (Inaba et al., 1994); however, we appeal to heuristics to accelerate the process even close to a linear order of complexity in n under mild conditions (Kanungo et al., 2002). Other approaches to implement the X-means algorithm faster are discussed by Pelleg and Moore (2000). Fitting a mixture of u normal distributions requires only $\mathcal{O}(nu^2)$ flops (Verbeek et al., 2003). Using the kernel trick does not impact the complexity of the algorithm, and moreover, the number of iterations is $\mathcal{O}(\log n)$. Hence, from a computational complexity perspective, the algorithm is dominated by (26) which can be solved using an LU decomposition in $\frac{2}{3}n^3 + \mathcal{O}(n^2)$ operations (Trefethen and Bau III, 1997). As such, the complexity of the algorithm is $\mathcal{O}(n^3 \log n)$.

4. Experiments

In this section, we apply the proposed ADG algorithm to a number of data sets, both real and artificial, and compare it to five alternative methods. Three of the methods in comparison are the Cost-Sensitive Support Vector Machine (CS-SVM), Synthetic Minority Over-Sampling Technique (SMOTE) (Chawla et al., 2002) and Borderline-SMOTE (BSMOTE) (Han et al., 2005). CS-SVM is an SVM algorithm (Hastie et al., 2009) modified for the imbalanced classification by imposing a higher cost on minority miss-classification (Elkan, 2001). In CS-SVM, we choose the value of the so-called "box constraint" in SVM to be

$\frac{l+l_-}{2l}$ for positive samples and $\frac{l+l_-}{2l_-}$ for negative samples, so that the cost ratio for the two-class misclassification is $\frac{l}{l_-}$. BSMOTE is similar to SMOTE but generates the new data points close to the boundary between the minority and majority classes. In this regard, BSMOTE uses a data synthesizing mechanism closest to that used in the proposed ADG. For both SMOTE and BSMOTE, once the new data points are generated, we can use a KFD algorithm to perform the task of classification on the balanced data set. Thereby, it would be straightforward to compare their performance against the ADG, as ADG also has the KFD as its classifier. The main parameter in SMOTE and B-SMOTE is the amount of oversampling, which is set to the same level as that in ADG, which, in turn, is determined by the value of q as discussed in Section 3.

The aforementioned competing algorithms are selected to compare different data generating mechanisms (SMOTE and BSMOTE) with that of the ADG, and to observe how they perform compared to another school of thought in imbalanced classification, cost-sensitive classification (CSSVM). We therefore present comparison among these algorithms in more detail. As a general principle, we select the parameters in the competing methods based on the recommendations made by the authors of the associated papers, unless otherwise indicated.

To further investigate ADG's viability as a means for imbalanced classification, at the end of this section we also compare the results of ADG with a combination of ensemble learners and undersampling (Wallace et al., 2011), and generating data using a fitted probabilistic distribution for the minority data points (Hempstalk et al., 2008; Liu et al., 2007). The former, referred to as "Under+ENS" hereafter, undersamples the majority data points several times to obtain balanced data sets and then uses a set of ensemble classifiers on the balanced data sets. The latter, referred to as "Prob-Fit" hereafter, fits a probability distribution to the existing minority data points and then generates synthetic data points from that distribution to create balanced data sets which, in turn, are used for classification. The probability distribution used in Prob-Fit, for all the data sets used in this paper, is a mixture of Gaussian distributions.

Concerning the kernel function used in both ADG and SVM (recall SVM is used in CS-SVM, SMOTE and BSMOTE), we use a Radial Basis Function kernel $K(\boldsymbol{x}, \boldsymbol{y}) = \exp(-d\|\boldsymbol{x} - \boldsymbol{y}\|^2)$, in which the parameter d is estimated through cross validation. To implement KFD we use the MATLAB package `Statistical Pattern Recognition Tool (STPRtool)` (Franc, 2011). We code ADG, SMOTE, BSMOTE, Under+ENS, and Prob-Fit in MATLAB, and also use the SVM implementation in MATLAB.

The performance measures we are interested in are the false alarm rate and detection power. Specifically, for the test set $\{(\boldsymbol{x}_\ell, y_\ell) | \ell = 1, 2, \ldots, N\}$, we can estimate the false alarm rate and detection power as follows

$$\widehat{\mathrm{FA}} = \frac{1}{N_-} \sum_{\ell=1}^{N_-} \mathcal{L}_{(0,1)}(y_\ell, \hat{y}_\ell), \quad \text{for } \ell \,\text{such that}\, y_\ell = -1, \tag{34}$$

and

$$\widehat{\mathrm{DP}} = 1 - \frac{1}{N_+} \sum_{\ell=1}^{N_+} \mathcal{L}_{(0,1)}(y_\ell, \hat{y}_\ell), \quad \text{for } \ell \,\text{such that}\, y_\ell = +1, \tag{35}$$

where N_- and N_+ are the number of majority and minority points in the test set, respectively. The variable $\hat{y}_i$ is the predicted class label (i.e. -1 or 1) for the associated prediction method, and $\mathcal{L}_{(0,1)}(.,.)$ is the 0-1 loss function

$$\mathcal{L}_{(0,1)}(y_1, y_2) = \begin{cases} 0 & \text{if } y_1 = y_2, \\ 1 & \text{if } y_1 \neq y_2. \end{cases} \tag{36}$$

Concerning the numerical experiments, we need to utilize simulated/real data sets which are deemed imbalanced. However, the number of available imbalanced data sets is limited, and we are also interested in testing algorithms on data sets with varying degrees of imbalance ratio, which can be characterized by the proportion of the majority data points to the minority data points in each data set. To this end, having the original training sets, $\mathcal{X}^+$ and $\mathcal{X}^-$, we can build training sets that are comprised of a subset of $\mathcal{X}^+$ and $\mathcal{X}^-$ and have a different proportion of majority to minority compared to the original training sets. That is, we have $\mathcal{X}_u^+ \subset \mathcal{X}^+$ and $\mathcal{X}_u^- \subset \mathcal{X}^-$ where $\frac{\mathcal{X}_u^+}{\mathcal{X}_u^-} > \frac{\mathcal{X}^+}{\mathcal{X}^-}$. Then we can utilize $\mathcal{X}_u^+$ and $\mathcal{X}_u^-$ as the new training set and the remaining data for testing. We will explain this approach in Section 4.2.

4.1 Using a Simulated Data set

Before presenting the classification results using the real data sets, we want to observe the difference of the mechanism of data generation between ADG and SMOTE. For this purpose, we create one simulated data set, in which we generate 900 data points as the majority data set from a mixture of five Gaussian distributions on $\mathbb{R}^2$ and 450 data points as the minority data set from a mixture of another five Gaussian distributions on $\mathbb{R}^2$.

Figure 1 shows a sample of synthetic data generation for a subset of the mixture of Gaussian distributions with an imbalance ratio greater than 6. Comparing region A in plots (b) and (c) in Figure 1 suggests that, for this particular data set, the ADG mechanism is more "space-filling" than that of SMOTE. Comparing region B in plots (b) and (d) shows that the intrusion into the majority space, while attempting to be space-filling, is less of a problem for ADG than that for BSMOTE, which also aims at generating data close to the boundary. Performing this space-filling property within the minority region, is of paramount importance for imbalanced classification in higher dimensions as well. It is not easy to demonstrate this property for the other data sets, as their dimensions are larger than two. The subsequent numerical results, however, support ADG's potency in imbalanced classification, and we think its strength can be partly attributed to ADG's ability to maintain the property better than SMOTE and BSMOTE.

4.2 Real Data sets

We use a total of eleven real data sets for training and testing. Four of them are from the UCI Machine Learning Repository (`http://archive.ics.uci.edu/ml/`), which are the Wisconsin Diagnostic Breast Cancer data set, the Ionosphere data set, the Yeast data set and Speech Recognition data set. The other seven are used in (Wallace and Dahabreh, 2012) (`http://www.cebm.brown.edu/static/imbalanced-datasets.zip`). Table 1 summarizes the basic properties associated with these data sets, including the Gaussian mixture data simulated in Section 4.1.

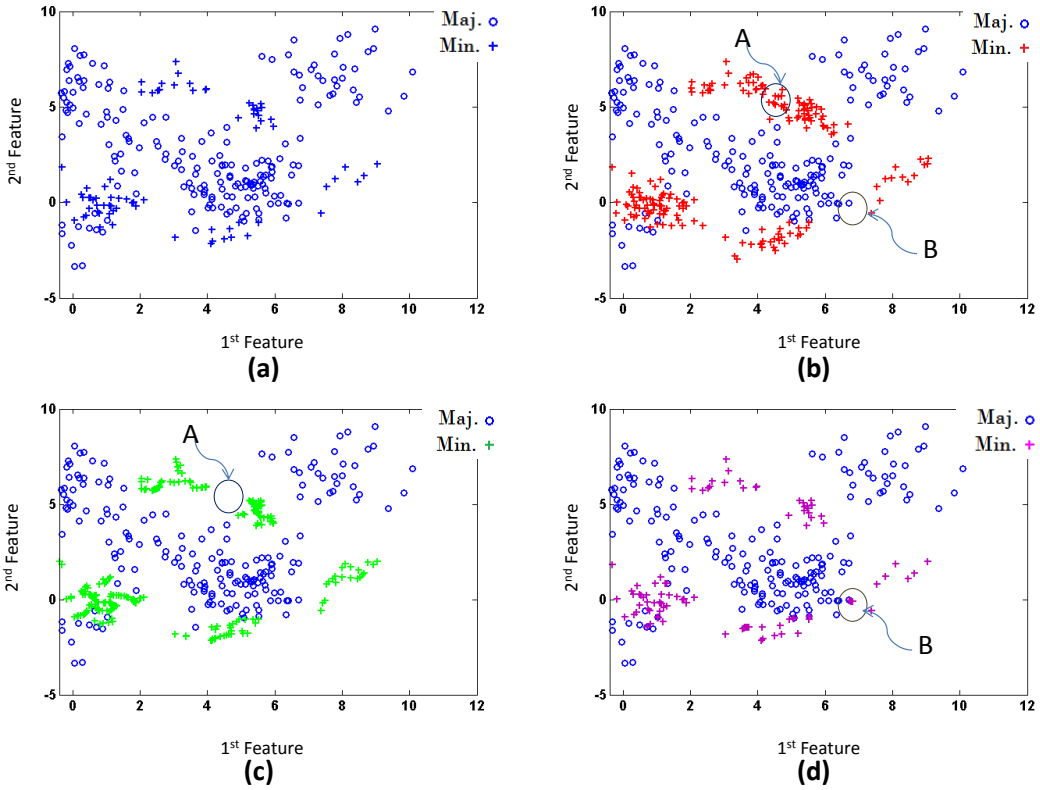


Figure 1: Comparing the mechanism of data generation in ADG with SMOTE for an artificial data set: (a) Original imbalanced data; (b) Balanced data after one iteration of ADG; (c) Balanced data after using SMOTE; (d) Balanced data after using BSMOTE. Comparing region A in plots (b) and (c) and region B in plots (c) and (d) shows ADG is more space-filling and intrudes less into the majority space.

Among the aforementioned data sets, not all of them are genuinely imbalanced. In those circumstances, we form the training data sets using a large portion of the majority data and a very small portion of the minority data. Besides, we are interested in observing how different methods perform as a data set becomes more imbalanced. For this purpose, we adjust the degrees of imbalance in a training set, by tuning the ratio of the number of majority points over the number of minority points in the data set. Specifically, for a given imbalance ratio, we first randomly undersample both the majority and the minority data points so that the training data set is constructed with the specified degree of imbalance. This means we obtain new training sets $\mathcal{X}_u^+$ and $\mathcal{X}_u^-$ as explained in the beginning of this section, run each algorithm on the training set, and use the remaining data for testing. We repeat this procedure ten times and report the average values as the estimated false alarm rate and detection power. Note that these new $\mathcal{X}_u^+$ and $\mathcal{X}_u^-$ will have the role of $\mathcal{X}^+$ and $\mathcal{X}^-$ in Algorithm 1 and no further modification is applied to the algorithm.

Data set	Dimension	Total Data Amount	# of Majority	# of Minority
Simulated Gaussian mixtures	2	1350	900	450
Breast Cancer Detection	9	699	458	241
Speech Recognition	10	990	900	90
Yeast	10	1484	1449	35
Ionosphere	34	351	225	126
Pima	8	768	500	268
Car	21	1728	1659	69
Ecoli	9	336	301	35
Glass	9	214	197	17
Haberman	3	306	225	81
Vehicle	18	846	634	212
CMC	24	1473	1140	333

Table 1: Basic properties of data sets

4.3 Results

We represent the performance of each algorithm on each data set using the Area Under Curve (AUC) of the Receiver Operating Characteristic (ROC) plot (Bradley, 1997). In the ROC analysis we plot each (FA, DP) point for a test case in an ROC space in which the FA is on the x-axis and the DP is on the y-axis (Provost et al., 1997). We use the `perfcurve` command in MATLAB to generate the ROC curves, once we have computed a sufficient number of (FA, DP) points. Then, we compute AUC as the area under a respective ROC curve. Note that a larger AUC generally denotes better performance.

We apply the six competing methods (ADG included) to the twelve data sets (including the simulated Gaussian mixture data) under different imbalance ratios. We report the average AUC and its standard deviation (both from ten repetitions), instead of the ROC plots themselves. Considering the number of classification methods in comparison, data sets involved, and imbalance ratios used, it is impractical to hope that plotting all ROC curves can produce a clear overall picture. Instead, we present the AUC information in a concise form: Table 2 lists the average values and Table 3 lists the corresponding standard deviations.

As evident in Table 2, ADG provides the largest AUC for most cases, especially under the most imbalanced circumstances of each test instance. Rather than expecting the ROC to suggest the optimal classifier, one may identify the regions or scenarios where a classifier can be recommended (Provost et al., 1997). We find that ADG provides a good balance between the conflicting objectives of reducing the false alarm, while increasing the detection power.

As expected, Prob-Fit performs very well on the simulated data, because the data are simulated using Gaussian mixture models. On the real data sets, the performance of Prob-Fit depends on the actual number of minority data points, that is, it performs better when the minority data are enough to reliably fit a distribution, and it performs poorly when the data set suffers from absolute scarcity. Therefore, simply fitting a distribution to generate data is of little use (Liu et al., 2007). The mechanism behind the performance of Under+ENS seems to be more involved, and it appears to be competitive for a few cases only. The comparisons demonstrate the importance of the structure of specific data sets,

and that no one classifier is dominant for all types of data under all imbalance ratios. The relation between a data structure and the mechanism embedded in the classifiers to handle the imbalanced data is of interest to be understood, but currently there are not enough insights garnered and we leave that issue to future efforts.

The fact that there are no dominating classifiers leads us to ask whether ADG's performance is statistically significant compared to the other methods. Considering that we are in presence of several classifiers and several data sets, we need to use a test which ranks classifiers based on their performance, followed by a post hoc analysis. One classical method which we utilize is the Friedman test (Demšar, 2006), a non-parametric method which sorts the algorithms conducted on several data sets. Let m^a be the number of algorithms, i.e. classifiers, and m^d be the number of data sets. Let Re be an $m^d \times m^a$ matrix of the results listed in Table 2, in which each row represents a data set and each column is a classifier. Considering the average results for each imbalance ratio as produced by one "data set", we have $m^d = 48$ and $m^a = 6$. First, define the matrix Ra whose entries in each row represent the classifier's rank for that specific data set. Under the null hypothesis that all classifiers are equivalent, i.e. their performance on each data set is identical, the Friedman statistic

$$\mathcal{F} = \frac{12m^d}{m^a(m^a+1)} \left(\sum_{\ell=1}^{m^a} \overline{Ra_\ell}^2 - \frac{m^a(m^a+1)^2}{4} \right), \tag{37}$$

has a Chi-squared distribution with $m^a - 1$ degrees of freedom, where $\overline{Ra_\ell}$ is the average value of column $\ell = 1, 2, \ldots, m^a$. Table 4 lists the means for the estimated ranks associated with each method. Figure 2, which presents the post hoc analysis on the ranking data using multiple comparisons, shows the ADG's ranking is significantly higher than other competing algorithms under the 0.05 level of significance.

Before concluding this section, we want to briefly discuss the drawbacks of the cost-sensitive approach (Maloof, 2003) and one-class classification (also known as novelty detection) (Park et al., 2010). One major obstacle faced with cost-sensitive methods is how to choose a suitable cost ratio that leads to robust outcomes. Figure 3 shows the detection power and false alarm as a function of cost ratio for the Haberman data where an imbalance ratio greater than 3 is used in training. Specifically, the cost ratio denotes the value associated with the box constraint in the SVM for minority data points divided into that value for the majority data points. As Figure 3 shows, the detection power remains almost constant after the cost ratio passes a threshold around 7, yet the false alarm rate continues to increase. Similar evidence has been documented in the literature regarding the lack of robustness in choosing a good cost ratio in the cost-sensitive methods (Byon et al., 2010). This lack of robust performance is one reason why synthetic oversampling is generally more powerful than cost-sensitive methods.

Some researchers favor one-class classification (OCC) approaches to solve imbalanced data problems. In other words, it is better to ignore the data points due to their sparseness in the minority data set, and instead create a closed decision boundary to characterize the majority data only. In a detection mission, one would classify a new data point as belonging either to the majority or the minority class. This OCC approach can be useful for some extreme cases in which the number of data points in the minority is so few that there are no practical ways to elicit any relevant information. In many practical cases, however, despite

Data	Imb. Ratio	ADG	SMOTE	BSMOTE	CSSVM	Under+ENS	Prob-Fit
	7	**0.886**	0.879	0.879	0.886	0.601	0.879
	4	0.888	0.886	0.881	0.888	0.675	**0.903**
Gaussian Mixture	3	0.885	0.887	0.878	0.890	0.659	**0.912**
	2	0.892	0.900	0.886	0.893	0.666	**0.906**
	6	**0.900**	0.896	0.897	0.895	0.814	0.882
	4	**0.899**	0.893	0.894	0.894	0.856	0.889
Breast Cancer	3	**0.905**	0.901	0.902	0.899	0.879	0.900
	2	0.899	0.897	0.897	0.894	0.903	**0.916**
	29	**0.894**	0.877	0.868	0.871	0.663	0.860
	15	**0.911**	0.900	0.908	0.906	0.774	0.902
Speech Recognition	10	0.891	0.898	0.903	0.891	0.867	**0.915**
	7	0.925	0.919	0.921	0.897	**0.932**	0.909
	121	**0.811**	0.709	0.731	0.760	0.614	0.778
	65	**0.820**	0.723	0.755	0.775	0.683	0.789
Yeast	40	**0.849**	0.766	0.812	0.801	0.765	0.810
	27	0.858	0.780	0.807	0.809	0.825	**0.859**
	6	**0.896**	0.890	0.884	0.891	0.796	0.854
	4	**0.891**	0.885	0.878	**0.891**	0.841	**0.891**
Ionosphere	3	0.895	0.888	0.881	0.894	0.869	**0.905**
	2	0.899	0.892	0.885	0.893	0.906	**0.918**
	6	0.681	0.622	0.679	0.668	0.680	**0.718**
	4	0.710	0.660	0.697	0.692	0.702	**0.729**
Pima	3	**0.721**	0.687	0.699	0.692	0.709	0.720
	2	0.734	**0.753**	0.724	0.700	0.709	0.736
	69	**0.890**	0.872	0.875	0.889	0.851	0.597
	37	0.898	0.888	0.891	0.896	**0.917**	0.756
Car	23	0.900	0.895	0.897	0.899	**0.970**	0.873
	15	0.904	0.897	0.903	0.903	**0.991**	0.900
	25	**0.729**	0.641	0.696	0.619	0.724	0.681
	14	0.732	0.616	0.705	0.601	**0.773**	0.697
Ecoli	8	0.731	0.702	0.701	0.613	**0.849**	0.715
	6	0.752	**0.797**	0.681	0.699	0.775	0.722
	33	0.713	0.653	0.669	**0.718**	0.667	0.710
	19	**0.754**	0.716	0.663	0.709	0.649	0.693
Glass	11	**0.779**	0.737	0.768	0.728	0.701	0.729
	8	0.826	**0.896**	0.852	0.796	0.808	0.774
	8	0.602	0.568	0.549	0.518	0.595	**0.608**
	4	**0.640**	0.543	0.584	0.569	0.586	0.601
Haberman	3	**0.653**	0.582	0.573	0.598	0.605	0.625
	2	**0.681**	0.596	0.584	0.596	0.618	0.627
	9	**0.714**	0.693	0.708	0.712	0.700	0.701
	5	**0.729**	0.707	0.728	0.729	0.701	0.709
Vehicle	3	0.783	0.778	0.763	**0.831**	0.712	0.729
	2	0.782	0.796	0.790	**0.843**	0.770	0.735
	10	0.589	0.532	0.538	0.586	**0.607**	0.549
	5	**0.679**	0.593	0.593	0.664	0.639	0.555
CMC	3	0.682	0.646	0.667	**0.712**	0.652	0.605
	2	0.692	0.683	0.678	**0.727**	0.670	0.641

Table 2: Average Area Under Curve (AUC). The largest values in each row are boldfaced. "Imb. Ratio" means imbalance ratio, the ratio of the number of majority points over the number of minority points in a data set.

Data	Imb. Ratio	ADG	SMOTE	BSMOTE	CSSVM	Under+ENS	Prob-Fit
	7	0.053	0.034	0.037	0.032	0.061	0.012
	4	0.065	0.037	0.037	0.037	0.061	0.008
Gaussian Mixture	3	0.048	0.047	0.050	0.048	0.050	0.008
	2	0.020	0.016	0.017	0.024	0.023	0.009
	6	0.008	0.013	0.012	0.009	0.009	0.015
	4	0.015	0.011	0.012	0.013	0.015	0.008
Breast Cancer	3	0.011	0.010	0.010	0.012	0.012	0.010
	2	0.013	0.011	0.020	0.013	0.012	0.009
	29	0.021	0.021	0.018	0.018	0.021	0.018
	15	0.044	0.033	0.029	0.033	0.041	0.021
Speech Recognition	10	0.027	0.042	0.020	0.034	0.031	0.010
	7	0.035	0.020	0.023	0.032	0.033	0.012
	121	0.028	0.039	0.045	0.029	0.029	0.046
	65	0.042	0.055	0.044	0.032	0.039	0.046
Yeast	40	0.070	0.075	0.067	0.063	0.073	0.069
	27	0.165	0.163	0.154	0.135	0.165	0.152
	6	0.038	0.032	0.030	0.036	0.037	0.028
	4	0.032	0.029	0.027	0.034	0.039	0.030
Ionosphere	3	0.028	0.031	0.020	0.029	0.028	0.013
	2	0.024	0.023	0.022	0.019	0.023	0.022
	6	0.026	0.022	0.021	0.031	0.030	0.017
	4	0.031	0.037	0.023	0.034	0.033	0.016
Pima	3	0.019	0.020	0.021	0.021	0.023	0.018
	2	0.026	0.023	0.027	0.028	0.027	0.022
	69	0.033	0.040	0.050	0.033	0.033	0.054
	37	0.025	0.074	0.068	0.031	0.028	0.120
Car	23	0.015	0.024	0.028	0.015	0.016	0.037
	15	0.006	0.040	0.043	0.005	0.006	0.082
	25	0.060	0.082	0.073	0.070	0.070	0.089
	14	0.075	0.091	0.087	0.073	0.090	0.085
Ecoli	8	0.045	0.051	0.049	0.039	0.047	0.058
	6	0.154	0.144	0.138	0.140	0.144	0.130
	33	0.083	0.094	0.096	0.088	0.098	0.092
	19	0.114	0.118	0.114	0.108	0.134	0.109
Glass	11	0.150	0.174	0.113	0.149	0.155	0.126
	8	0.146	0.138	0.156	0.132	0.161	0.133
	8	0.041	0.040	0.040	0.042	0.043	0.037
	4	0.053	0.057	0.043	0.049	0.060	0.029
Haberman	3	0.045	0.055	0.064	0.046	0.053	0.054
	2	0.049	0.053	0.053	0.043	0.049	0.050
	9	0.016	0.019	0.017	0.017	0.019	0.017
	5	0.025	0.027	0.026	0.029	0.028	0.027
Vehicle	3	0.027	0.024	0.024	0.029	0.029	0.019
	2	0.033	0.051	0.042	0.027	0.031	0.054
	10	0.026	0.048	0.057	0.023	0.025	0.080
	5	0.016	0.038	0.049	0.016	0.019	0.060
CMC	3	0.020	0.021	0.022	0.020	0.024	0.022
	2	0.073	0.089	0.076	0.079	0.079	0.088

Table 3: Standard deviation for Area Under Curve (AUC) reported in Table 2.

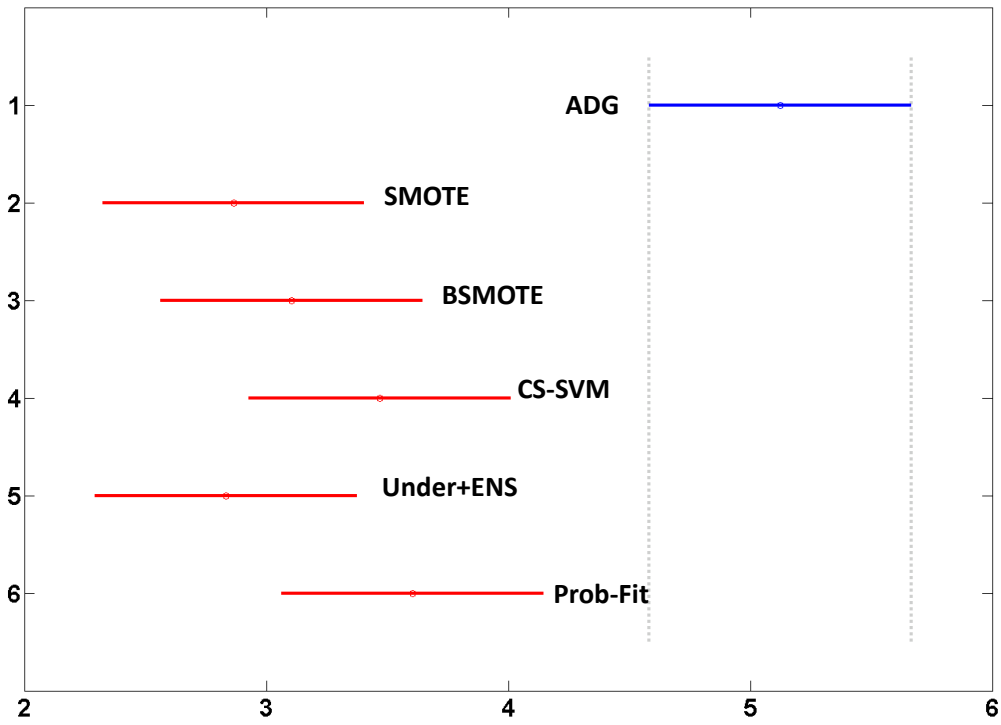


Figure 2: Post hoc analysis on the ranking data obtained by the Friedman test. ADG's mean column rank is significantly higher than other classifiers.

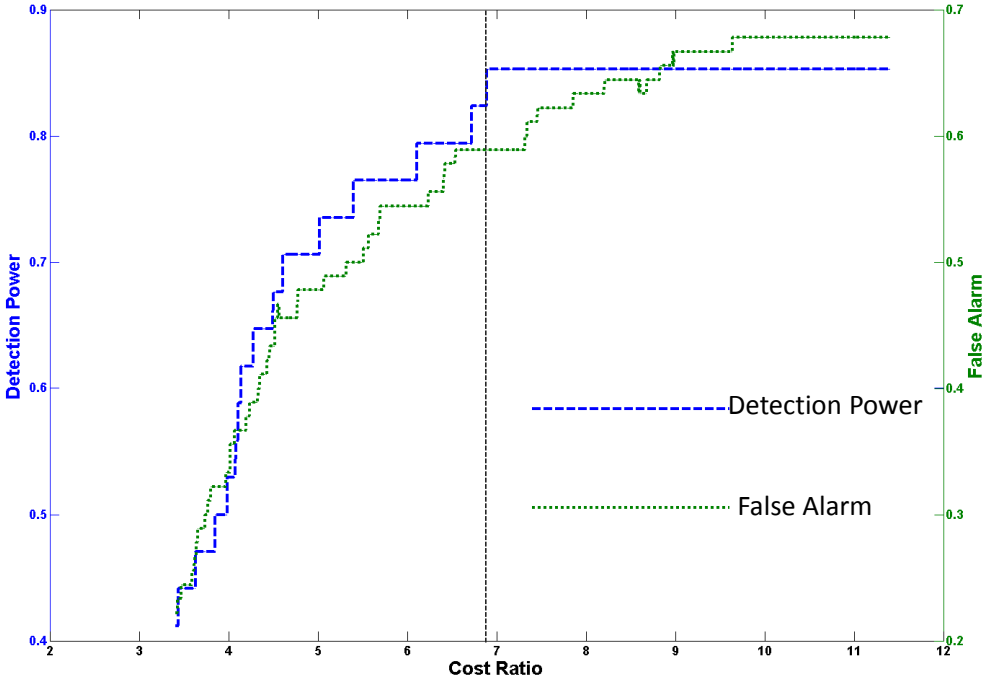


Figure 3: Detection power (left axis) and false alarm (right axis) as a function of the cost ratio in CS-SVM for the Haberman data set.

Classifier	ADG	SMOTE	BSMOTE	CSSVM	Under+ENS	Prob-Fit
Mean of Ranking	5.125	2.865	3.104	3.469	2.833	3.604

Table 4: Mean of rankings based on Friedman test

the sparseness of the data, minority data sets still can provide useful information if utilized appropriately. To demonstrate the usefulness of utilizing the minority data, we compare ADG with the OCC method developed in Park et al. (2010) using four sample data sets; this OCC method was proven to provide asymptotically the tightest bound for majority data points. For these four sample data sets, we select the training and test data such that the training data sets have the smallest value of imbalance ratio reported in Table 2. As Figure 4 shows, the OCC could be effective, for instance, duplicating ADG's performance in the case of Pima data. One drawback is that OCC methods often suffer from a high false alarm rate, while attaining a high detection power (e.g. in the case of the Ionosphere data). When an OCC tries to build the tightest possible closed boundary around the majority data, the result can be an over-tightened boundary, instead of a boundary loose enough to identify all majority data points. On the other hand, in the two-class cases, the existence of minority data points can actually help relax the position of the decision boundary, at least locally where these minority data points are present. For more detailed comparisons of another OCC method with two-class classifiers, the reader may consult (Hempstalk et al., 2008); the results presented there also confirm the argument that if minority data are utilized, one generally observes an improvement in the minority detection.

5. Extension and Error Bounds

In this section, we consider two additional aspects regarding the proposed algorithm. First, we seek to identify bounds on the generalization error for the ADG. Second, we extend the proposed method to deal with the multi-class classification in which a subset of classes has very few observations available in the training stage.

5.1 Bounds on Generalization Error

Generalization error refers to the expected error on test instances coming from the same distribution of the training sample (Rasmussen and Williams, 2006). Specifically, if $\boldsymbol{x} \sim \mathcal{G}$, where $\mathcal{G}$ is the distribution of the input $\boldsymbol{x}$, the generalization error of some decision function h with respect to loss function $\mathcal{L}$ is defined as

$$\mathbb{E}_{\boldsymbol{x}}\{\mathcal{L}(h)\}, \tag{38}$$

where $\mathbb{E}$ is the expectation operator.

Let $\boldsymbol{\alpha}_F$ denote the optimal value of $\boldsymbol{\alpha}$ obtained by solving optimization problem (3), namely the KFD. Similar to the procedure explained in Section 3 for obtaining the prediction label for ADG, let $C_{\mathcal{U}}$ be the same one-dimensional binary classifier used for ADG, trained on the set $\mathcal{U} = \{(h(\boldsymbol{x}_\ell; \boldsymbol{\alpha}_F), y_\ell) : \boldsymbol{x}_\ell \in \mathcal{X}^- \cup \mathcal{X}^+, \ell = 1, 2, \ldots, n\}$, where $\boldsymbol{\kappa}_{\boldsymbol{x}}$ is defined similarly to (32) for $\boldsymbol{x}_\ell \in \mathcal{X}^- \cup \mathcal{X}^+$, and $h(\boldsymbol{x}_\ell; \boldsymbol{\alpha}_F) = \boldsymbol{\alpha}_F^T \boldsymbol{\kappa}_{\boldsymbol{x}_\ell}$. If the threshold value for the

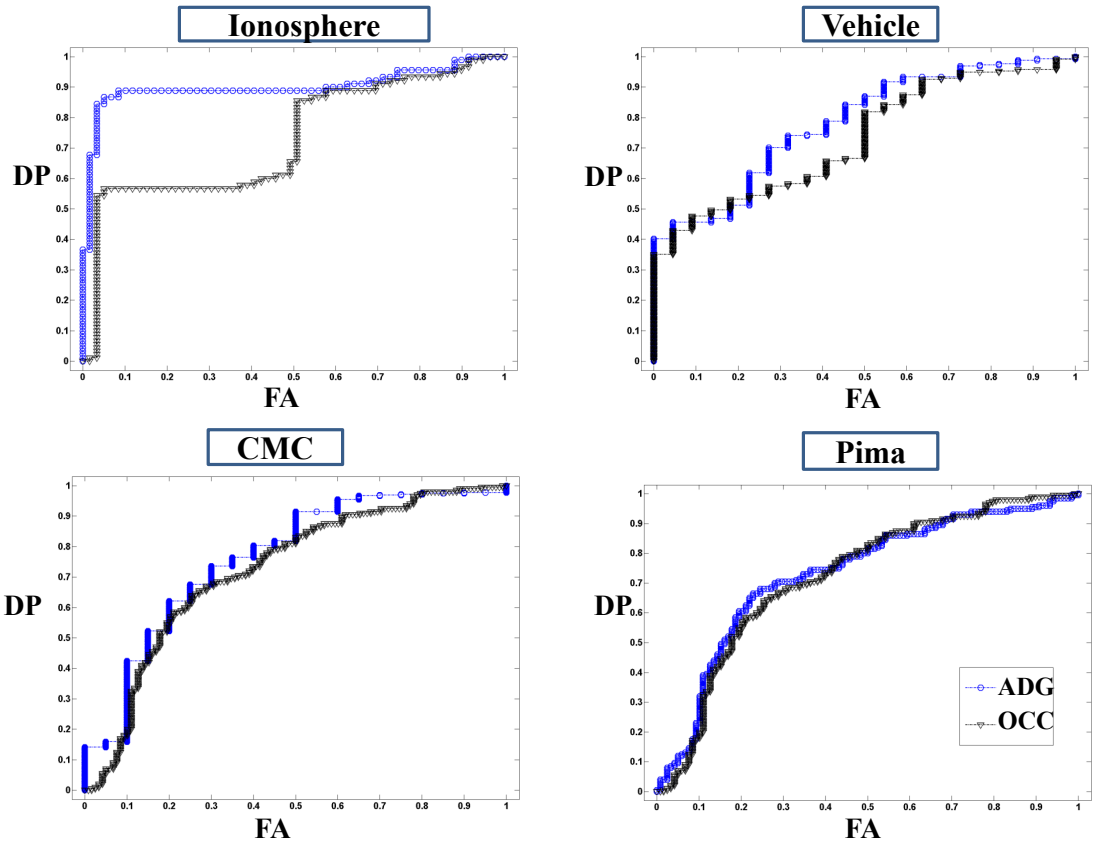


Figure 4: Comparing ROCs for ADG and OCC for four sample data sets.

$C_{\mathcal{U}}$ is v_F, we have the following prediction for a test point $\boldsymbol{x}_t$ using the KFD

$$\text{KFD}(\boldsymbol{x}_t) = \begin{cases} 1 & \text{if } h(\boldsymbol{x}_t; \boldsymbol{\alpha}_F) > v_F, \\ -1 & \text{if } h(\boldsymbol{x}_t; \boldsymbol{\alpha}_F) \leq v_F. \end{cases} \tag{39}$$

Consequently, following the total law of probability, we can deduce that the generalization error of KFD is equal to

$$err_{\text{K}} = \pi_- \mathbb{P}\left[h(\boldsymbol{x}_t; \boldsymbol{\alpha}_F) > v_F | y_t = -1\right] + \pi_+ \mathbb{P}\left[h(\boldsymbol{x}_t; \boldsymbol{\alpha}_F) \leq v_F | y_t = 1\right], \tag{40}$$

where π_i is the prior probability that a point belongs to the class $i \in \{-, +\}$.

Durrant and Kabán (2012) established an upper bound on this generalization error, under the assumption that the data points of each class follow a Gaussian distribution once mapped to the feature space. Specifically, having a training data set of size $n = l_+ + l_-$ and assuming data in the feature space are normally distributed with mean $\boldsymbol{\mu}_i$ and covariance matrix $\boldsymbol{\Sigma}$ for $i \in \{-, +\}$, then for any $\rho \in (0, 1)$ the generalization error of KFD is bounded above with probability of at least $1 - \rho$ by $ub(l, \rho)$ where

$$ub(l, \rho) = \sum_{i \in \{-,+\}} \pi_i \Phi\left(-2\left[g(\bar{\tau}(\epsilon)) \times \Pi - \sqrt{\frac{n}{l_i}}\left(1 + \sqrt{\frac{2}{n}\log\frac{4}{\rho}}\right)\right]\right), \tag{41}$$

where

$$\Pi = \left[\sqrt{\frac{\|\boldsymbol{\mu}_+ - \boldsymbol{\mu}_-\|^2}{\lambda\max(\boldsymbol{\Sigma})} + \frac{n}{l_- l_+}\frac{\text{tr}(\boldsymbol{\Sigma})}{\lambda\max(\boldsymbol{\Sigma})}} - \sqrt{\frac{2n}{l_- l_+}\log\frac{4}{\rho}}\right]_+, \tag{42}$$

$g(r) = \frac{\sqrt{r}}{1+r}$ for $r \in \mathbb{R}$, $\lambda\max(\boldsymbol{\Sigma})$ is the largest eigenvalue of the covariance matrix, $[.]_+ = \max(0, .)$, Φ is the CDF of the standard normal distribution, and

$$\bar{\tau}(\epsilon) = \frac{\lambda\max(\boldsymbol{\Sigma})}{\eta}\left(1 + \sqrt{\frac{n-2}{n}} + \frac{\epsilon}{\sqrt{n}}\right)^2 + \tau(\boldsymbol{\Sigma}_n), \tag{43}$$

where $\epsilon = \sqrt{2\log\frac{4}{\rho}}$, $\tau(\boldsymbol{\Sigma}_n)$ denotes the condition number of $\boldsymbol{\Sigma}_n$ that is the covariance matrix of the points in a subset of the feature space generated by the n points in $\mathcal{X}^- \cup \mathcal{X}^+$, and η is a regularization constant to ensure non-singularity of the estimate of $\boldsymbol{\Sigma}_n$. As $g(.)$ is a monotonic decreasing function on $r \geq 1$, a smaller value for $\bar{\tau}(\epsilon)$ suggests a smaller value for the upper bound. Note that assuming the regularization constant η does not need to change as more data points are added to the training set, then the only quantities which affect $\bar{\tau}(\epsilon)$ are $\tau(\boldsymbol{\Sigma}_n)$ and n.

Note that as the number of observations increases, (41) yields a tighter bound, assuming that all other quantities remain constant. This is in fact what happens in synthetic data generation, especially for ADG, since it generates extra observations at each iteration of the algorithm. The more subtle issue is how the estimated value of the covariance matrix $\boldsymbol{\Sigma}$, projected in the Hilbert space generated by the observation, changes with the generation of more data points.

Note that $\boldsymbol{\Sigma}_n = \boldsymbol{P}\boldsymbol{\Sigma}\boldsymbol{P}^T$, where $\boldsymbol{P}$ is an orthogonal projection into the Hilbert space spanned by the observations. Assuming that data points mapped to the feature space are

linearly independent, we can have $\boldsymbol{P}_n = \left(\boldsymbol{X}_n^{\boldsymbol{\phi}^T}\boldsymbol{X}_n^{\boldsymbol{\phi}}\right)^{-\frac{1}{2}}\boldsymbol{X}_n^{\boldsymbol{\phi}^T}$, where $\boldsymbol{X}_n^{\boldsymbol{\phi}}$ is a matrix whose columns are $\boldsymbol{\phi}(\boldsymbol{x}_\ell)$ for $\boldsymbol{x}_\ell \in \mathcal{X}^- \cup \mathcal{X}^+$. If we add a new observation $\boldsymbol{x}_{n+1}$ to the training set $\mathcal{X}^- \cup \mathcal{X}^+$, we will get the projection matrix $\boldsymbol{P}_{n+1}^{\boldsymbol{\phi}}$. See the Appendix for an explanation that as the number of data points increases, the condition number of the covariance matrix of the space generated by the data points in the feature space decreases, which in turn implies we achieve a tighter bound for generalization error using ADG.

In fact, as long as a synthetic data generation mechanism is embedded in a KFD framework, as ADG does, we can invoke the above theoretical result on the reduction of the generalization error. Despite the fact that SMOTE and BSMOTE can also be used, note that their data generation mechanisms cannot be integrated with KFD. For this reason, the above error bound result cannot be readily applied to SMOTE and BSMOTE.

5.2 Extension to Multi-class Classification

The methodology presented for the imbalanced two-class classification can be easily extended to cover multi-class classification in which a subset of classes lack sufficient observations for the training stage. Let $\mathcal{X}^i = \{\boldsymbol{x}_1^i, \boldsymbol{x}_2^i, \ldots, \boldsymbol{x}_{l_i}^i\} \subset \mathcal{X}$ denote the training set for class $i \in \mathcal{I} = \{1, 2, \ldots, I_s\}$, where $l_{i_2} \ll l_{i_1}$, for $i_1 \in \mathcal{I}_1$, $i_2 \in \mathcal{I}_2$, where $\mathcal{I}_1 \cup \mathcal{I}_2 = \mathcal{I}$ and $\mathcal{I}_1 \cap \mathcal{I}_2 = \emptyset$. Let $\mathcal{Z}_{i_s} = \{\boldsymbol{x}_{l_{i_s}+1}^{i_s}, \boldsymbol{x}_{l_{i_s}+2}^{i_s}, \ldots, \boldsymbol{x}_{l_{i_s}+k_{i_s}}^{i_s}\} \subset \mathcal{X}$ be the absent data from the minority class i_s, and denote each $\boldsymbol{x}_{l_{i_s}+k_{i_s}}$ by $\boldsymbol{z}_j^{i_s}$. For simplicity, consider a case in which the data in each group consist of a single cluster, i.e. $C = 1$; however, the following algorithm can be readily extended to consider more clusters. Assume that the data are centered around each covariate so they have mean 0. Sequentially solve the following optimization problem to obtain $\boldsymbol{w}_i$ for $i \in \mathcal{I}$:

$$\max_{\boldsymbol{w}_i} J(\boldsymbol{w}_i) = \frac{\boldsymbol{w}_i^T \boldsymbol{S}_B^{\boldsymbol{\phi}} \boldsymbol{w}_i}{\boldsymbol{w}_i^T \boldsymbol{S}_W^{\boldsymbol{\phi}} \boldsymbol{w}_i}, \tag{44}$$

subject to

$$\boldsymbol{w}_i \perp \boldsymbol{w}_\ell, \quad \forall \ell < i, \tag{45}$$

$$\left(\boldsymbol{w}_i^T \boldsymbol{\phi}(\boldsymbol{z}_j^{i_s}) - \boldsymbol{w}^T \boldsymbol{m}_{i_s}^{\boldsymbol{\phi}}\right)^2 \leq \delta, \tag{46}$$

$$(\boldsymbol{\phi}(\boldsymbol{z}_j^{i_s}) - \boldsymbol{m}_{i_d}^{\boldsymbol{\phi}})^T (\boldsymbol{\phi}(\boldsymbol{z}_j^{i_s}) - \boldsymbol{m}_{i_r}^{\boldsymbol{\phi}}) \leq \Lambda \quad \text{for} \quad j = 1, 2, \ldots k_{i_s}, \quad i_s \in \mathcal{I}_2, i_r \in \mathcal{I}_1, \tag{47}$$

where $\boldsymbol{S}_B^{\boldsymbol{\phi}}$ and $\boldsymbol{S}_W^{\boldsymbol{\phi}}$ are the between and within class scatter matrices, respectively, in the feature space

$$\boldsymbol{S}_B^{\boldsymbol{\phi}} = \sum_{i \in \mathcal{I}} l_i \boldsymbol{m}_i^{\boldsymbol{\phi}} (\boldsymbol{m}_i^{\boldsymbol{\phi}})^T,$$

$$\boldsymbol{S}_W^{\boldsymbol{\phi}} = \sum_{i \in \mathcal{I}} \sum_{\boldsymbol{x} \in \mathcal{X}^i} (\boldsymbol{\phi}(\boldsymbol{x}) - \boldsymbol{m}_i^{\boldsymbol{\phi}})(\boldsymbol{\phi}(\boldsymbol{x}) - \boldsymbol{m}_i^{\boldsymbol{\phi}})^T, \tag{48}$$

and $\boldsymbol{m}_i^{\boldsymbol{\phi}} = \frac{1}{l_i} \sum_{j=1}^{l_i} \boldsymbol{\phi}(\boldsymbol{x}_j^i)$, for $i \in \mathcal{I}$. For each minority class in $\mathcal{I}_2$, generate k_{i_s} artificial points from class $i_c \in \mathcal{I}_2$. Similar to the two-class classification problem, use the Representer's Theorem to replace each $\boldsymbol{w}_i$ and $\boldsymbol{\phi}(\boldsymbol{z}_j^{i_s}) - \boldsymbol{m}_{i_r}^{\boldsymbol{\phi}}$ as linear combinations of the training data in the feature space as in (7) and (8). This leads to systems of linear equations as in (26) which can be embedded into an algorithm similar to Algorithm 1.

6. Summary

This paper presents an algorithm for solving the two-class classification with imbalanced training data. The difficulty associated with such data structures is that the inadequate number of data points belonging to one class (i.e. minority) leads to the problem that most two-class classification algorithms tend to favor the majority class in labeling test points. To solve the problem, we devise an algorithm that relies on minority data synthesis. At each iteration we solve an optimization which considers more numbers of minority points without explicitly specifying them. Those points affect our decision by forcing the algorithm to set the decision boundary as though the points genuinely existed. We draw samples from the new region to enable a more accurate estimation for the scatter matrices. Using several simulated and real data sets, we compare the performance of the resulting ADG algorithm with the competing methods, CS-SVM, SMOTE, BSMOTE, Under+ENS and Prob-Fit. The results suggest that using ADG is preferable when there is a pronounced data imbalance.

This paper is a first step for developing a data mechanism embedded in a classification algorithm which we proved useful based on empirical evidence. Since the introduction of SMOTE (Chawla et al., 2002), there has been significant attention to synthetic data generation. We suggest however, that more research is needed to understand the relationship between data generation and classification algorithms.

There are a few critical issues which deserve further attention in this regard. First, the impact of the data structure on the data generation mechanism needs to be studied more thoroughly. The current procedure of data generation may not be suitable for all data structures. Certain alterations on the algorithm, based on the knowledge of how the physical system of interest works, can help improve the performance of ADG. Second, ADG can benefit from an investigation into certain assumptions made in the algorithm. One place is on the assumption that the absent data reside in existing clusters. While reasonable, it might be restrictive for some data sets. Another aspect is that in the current iteration of algorithm, we eliminate all artificial data points that fall on the majority side; this appears beneficial in the examples we studied. Whether or not it can be beneficial for all types of data remains unclear. These issues are certainly important and how to address them is an ongoing pursuit.

Acknowledgments

Arash Pourhabib, Yu Ding and Bani K. Mallick were partially supported by grants from NSF (DMS-0914951, CMMI-0926803, and CMMI-1000088) and King Abdullah University of Science and Technology (KUS-CI-016-04). The authors are also grateful of the valuable suggestions made by the editor and referees that greatly improved the paper.

Appendix A.

We want to show as the number of training data points increases, the condition number of the projected covariance matrix into the Hilbert space generated by the data points decreases. Let $\boldsymbol{x}_\ell \in \mathcal{X}$ for $\ell = 1, 2, \ldots, n$ denote the data points in the original space and let

$\boldsymbol{\zeta}_\ell$ for $\ell = 1, 2, \ldots, n$ denote the data points mapped to a separable Hilbert space $\mathcal{H}$ using a feature map $\boldsymbol{\phi}$, that is $\boldsymbol{\zeta}_\ell = \boldsymbol{\phi}(\boldsymbol{x}_\ell)$ for $\ell = 1, 2, \ldots, n$. Suppose $\mathcal{H}_n$ is an n dimensional subspace of $\mathcal{H}$ spanned by $\boldsymbol{\zeta}_\ell$ for $\ell = 1, 2, \ldots, n$. If $\boldsymbol{\zeta}_\ell$ follow a normal distribution in $\mathcal{H}$ with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, we can have $\boldsymbol{\Sigma}_n$ as the projected covariance matrix into the finite dimensional space $\mathcal{H}_n$. More precisely, $\boldsymbol{\Sigma}_n = \boldsymbol{P}_n \boldsymbol{\Sigma} \boldsymbol{P}_n^T$, namely $\boldsymbol{P}_n$ is an orthogonal projection into $\mathcal{H}_n$, where $\boldsymbol{P}_n = \left(\boldsymbol{X}_n^{\boldsymbol{\phi}^T} \boldsymbol{X}_n^{\boldsymbol{\phi}}\right)^{-\frac{1}{2}} \boldsymbol{X}_n^{\boldsymbol{\phi}^T}$ and $\boldsymbol{X}_n^{\boldsymbol{\phi}} = [\boldsymbol{\zeta}_\ell : \ell = 1, 2, \ldots, n]$. We want to show that the condition number of $\boldsymbol{\Sigma}_n$ is larger than or equal to that of $\boldsymbol{\Sigma}_{n+1}$.

Without loss of generality, after a rotation and scaling of the data, assume $\left(\boldsymbol{X}_p^{\boldsymbol{\phi}^T} \boldsymbol{X}_p^{\boldsymbol{\phi}}\right) = \boldsymbol{I}$, for $p \in \mathbb{N}$, where $\boldsymbol{I}$ is the identity matrix of appropriate size. Therefore,

$$\boldsymbol{P}_{n+1} = \left(\boldsymbol{X}_{n+1}^{\boldsymbol{\phi}}\right)^T = \left[(\boldsymbol{X}_n^{\boldsymbol{\phi}})^T | \boldsymbol{\zeta}_{n+1}^T\right], \tag{49}$$

and

$$\lambda_{\max}(\boldsymbol{\Sigma}_{n+1}) = \lambda_{\max}\left(\boldsymbol{P}_{n+1} \boldsymbol{\Sigma} \boldsymbol{P}_{n+1}^T\right) = \lambda_{\max}\left(\begin{bmatrix} (\boldsymbol{X}_n^{\boldsymbol{\phi}})^T \\ \boldsymbol{\zeta}_{n+1}^T \end{bmatrix} \boldsymbol{\Sigma} \left[\boldsymbol{X}_n^{\boldsymbol{\phi}} | \boldsymbol{\zeta}_{n+1}\right]\right) \tag{50}$$

$$= \lambda_{\max}\left(\begin{bmatrix} \boldsymbol{\Sigma}_n & (\boldsymbol{X}_n^{\boldsymbol{\phi}})^T \boldsymbol{\Sigma} \boldsymbol{\zeta}_{n+1} \\ \boldsymbol{\zeta}_{n+1}^T \boldsymbol{\Sigma} \boldsymbol{X}_n^{\boldsymbol{\phi}} & \boldsymbol{\zeta}_{n+1}^T \boldsymbol{\Sigma} \boldsymbol{\zeta}_{n+1} \end{bmatrix}\right). \tag{51}$$

Let $\|\boldsymbol{\zeta}_{n+1}\|^2 := \boldsymbol{\zeta}_{n+1}^T \boldsymbol{\Sigma} \boldsymbol{\zeta}_{n+1}$. Therefore,

$$\lambda_{\max}(\boldsymbol{\Sigma}_{n+1}) \leq \lambda_{\max}(\boldsymbol{\Sigma}_n) + \|\boldsymbol{\zeta}_{n+1}\|^2, \tag{52}$$

and

$$\lambda_{\min}(\boldsymbol{\Sigma}_{n+1}) \geq \lambda_{\min}(\boldsymbol{\Sigma}_n) + \|\boldsymbol{\zeta}_{n+1}\|^2. \tag{53}$$

Let $\tau(.)$ denote the condition number of a matrix, so

$$\tau(\boldsymbol{\Sigma}_{n+1}) = \frac{\lambda_{\max}(\boldsymbol{\Sigma}_{n+1})}{\lambda_{\min}(\boldsymbol{\Sigma}_{n+1})} \leq \frac{\lambda_{\max}(\boldsymbol{\Sigma}_n) + \|\boldsymbol{\zeta}_{n+1}\|^2}{\lambda_{\min}(\boldsymbol{\Sigma}_n) + \|\boldsymbol{\zeta}_{n+1}\|^2} < \tau(\boldsymbol{\Sigma}_n). \tag{54}$$

References

Rehan Akbani, Stephen Kwek, and Nathalie Japkowicz. Applying support vector machines to imbalanced datasets. In *Proceedings of the fifteenth European Conference on Machine Learning (ECML)*, pages 39–50. Springer, 2004.

Kurt Anstreicher and Henry Wolkowicz. On Lagrangian relaxation of quadratic matrix constraints. *SIAM Journal on Matrix Analysis and Applications*, 22(1):41–55, 1998.

Gustavo E.A.P.A. Batista, Ronaldo C. Prati, and Maria Carolina Monard. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explorations Newsletter-Special Issue on Learning from Imbalanced Datasets*, 6(1):20–29, 2004.

Peter J. Bickel and Elizaveta Levina. Some theory for Fishers linear discriminant function, naive Bayes, and some alternatives when there are many more variables than observations. *Bernoulli*, 10:989–1010, 2004.

Andrew P. Bradley. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159, 1997.

Eunshin Byon, Abhishek K. Shrivastava, and Yu Ding. A classification procedure for highly imbalanced class sizes. *IIE Transactions*, 42(4):288–303, 2010.

Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

J. J. Chen, C. A. Tsai, J. F. Young, and R. L. Kodell. Classification ensembles for unbalanced class sizes in predictive toxicology. *SAR and QSAR in Environmental Research*, 16(6):517–529, 2005.

Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7:1–30, 2006.

Robert J. Durrant and Ata Kabán. Error bounds for kernel Fisher linear discriminant in Gaussian Hilbert space. In *Proceedings of the fifteenth International Conference on Artificial Intelligence and Statistics (AISTATS 2012)*, volume 22, pages 337–345, 2012.

Charles Elkan. The foundations of cost-sensitive learning. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, pages 973–978, 2001.

Chris Fraley and Adrian E Raftery. How many clusters? Which clustering method? Answers via model-based cluster analysis. *The Computer Journal*, 41(8):578–588, 1998.

Vojtech Franc. *The Statistical Pattern Recognition Toolbox, Version 2.11.* 2011. URL `http://cmp.felk.cvut.cz/cmp/software/stprtool/index.html`.

Alexander Gammerman, Volodya Vovk, and Vladimir Vapnik. Learning by transduction. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, pages 148–155, 1998.

Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning. In *Advances in Intelligent Computing*, volume 3644 of *Lecture Notes in Computer Science*, pages 878–887. Springer Berlin Heidelberg, 2005.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition.* Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2009.

Haibo He and Edwardo A. Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, 2009.

Kathryn Hempstalk, Eibe Frank, and Ian H. Witten. One-class classification by combining density and class probability estimation. In Walter Daelemans, Bart Goethals, and Katharina Morik, editors, *Machine Learning and Knowledge Discovery in Databases*, volume 5211 of *Lecture Notes in Computer Science*, pages 505–519. Springer Berlin Heidelberg, 2008.

Thomas Hofmann, Bernhard Schölkopf, and Alexander J. Smola. Kernel methods in machine learning. *Annals of Statistics*, 36(3):1171–1220, 2008.

Mary Inaba, Naoki Katoh, and Hiroshi Imai. Applications of weighted Voronoi diagrams and randomization to variance-based k-clustering. In *Proceedings of the Tenth Annual Symposium on Computational Geometry*, pages 332–339. ACM, 1994.

Nathalie Japkowicz. The class imbalance problem: Significance and strategies. In *Proceedings of the 2000 International Conference on Artificial Intelligence (ICAI)*, pages 111–117, 2000.

Tapas Kanungo, David M Mount, Nathan S. Netanyahu, Christine D. Piatko, Ruth Silverman, and Angela Y. Wu. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24 (7):881–892, 2002.

Alexander Liu, Joydeep Ghosh, and Cheryl E. Martin. Generative oversampling for mining imbalanced datasets. In *International Conference on Data Mining*, pages 66–72, 2007.

Xu-Ying Liu, Jianxin Wu, and Zhi-Hua Zhou. Exploratory undersampling for class-imbalance learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 39(2):539–550, 2009.

Marcus A. Maloof. Learning when data sets are imbalanced and when costs are unequal and unknown. In *Proceedings of ICML-2003 Workshop on Learning from Imbalanced Data Sets II*, 2003.

Sebastian Mika, Gunnar Rätsch, Jason Weston, Bernhard Schölkopf, and Klaus-Robert Müllers. Fisher discriminant analysis with kernels. In *Neural Networks for Signal Processing IX, 1999. Proceedings of the 1999 IEEE Signal Processing Society Workshop.*, pages 41 –48, Aug 1999.

Chiwoo Park, Jianhua Z. Huang, and Yu Ding. A computable plug-in estimator of minimum volume sets for novelty detection. *Operations Research*, 58(5):1469–1480, Sep 2010.

Dan Pelleg and Andrew Moore. X-means: Extending K-means with efficient estimation of the number of clusters. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 727–734. Morgan Kaufmann, 2000.

Foster Provost, Tom Fawcett, and Ron Kohavi. The case against accuracy estimation for comparing induction algorithms. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 445–453. Morgan Kaufmann, 1997.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

Kai Ming Ting. An instance-weighting method to induce cost-sensitive trees. *IEEE Transactions on Knowledge and Data Engineering*, 14(3):659–665, 2002.

Lloyd N. Trefethen and David Bau III. *Numerical Linear Algebra*. SIAM: Society for Industrial and Applied Mathematics, 1997.

Jakob J. Verbeek, Nikos Vlassis, and Ben Kröse. Efficient greedy learning of Gaussian mixture models. *Neural Computation*, 15(2):469–485, 2003.

Byron C. Wallace and Issa J. Dahabreh. Class probability estimates are unreliable for imbalanced data (and how to fix them). In *IEEE Twelfth International Conference on Data Mining (ICDM)*, pages 695–704, 2012.

Byron C. Wallace, Kevin Small, Carla E. Brodley, and Thomas A. Trikalinos. Class imbalance, redux. In *IEEE Eleventh International Conference on Data Mining (ICDM)*, pages 754–763, 2011.

Gary M. Weiss. Mining with rarity: a unifying framework. *ACM SIGKDD Explorations Newsletter-Special Issue on Learning from Imbalanced Datasets*, 6(1):7–19, 2004.

Journal of Machine Learning Research 16 (2015) 2725-2749 Submitted 5/14; Revised 1/15; Published 12/15

Decision Boundary for Discrete Bayesian Network Classifiers

Gherardo Varando GHERARDO.VARANDO@UPM.ES
Concha Bielza MCBIELZA@FI.UPM.ES
Pedro Larrañaga PEDRO.LARRANAGA@FI.UPM.ES
Departamento de Inteligencia Artificial
Universidad Politécnica de Madrid
Campus de Montegancedo, s/n
28660 Boadilla del Monte, Madrid, Spain

Editor: Max Chickering

Abstract

Bayesian network classifiers are a powerful machine learning tool. In order to evaluate the expressive power of these models, we compute families of polynomials that sign-represent decision functions induced by Bayesian network classifiers. We prove that those families are linear combinations of products of Lagrange basis polynomials. In absence of V-structures in the predictor sub-graph, we are also able to prove that this family of polynomials does indeed characterize the specific classifier considered. We then use this representation to bound the number of decision functions representable by Bayesian network classifiers with a given structure.

Keywords: Bayesian networks, supervised classification, decision boundary, polynomial threshold function, Lagrange basis

1. Introduction

One of the problems with any supervised classification model, and Bayesian network classifiers in particular, is to understand the limits of the expressive power of these models. The first rigorous result in this direction was reported by Minsky (1961), showing that the decision boundary in naive Bayes classifiers with binary predictors is a hyperplane. Since then several other researchers have addressed the problem. Peot (1996) reviewed Minsky's results about binary predictors and presented some extensions. He mainly discussed the case of naive Bayes with k-valued observations and observation-observation dependencies. He also reported an upper bound on the number of linearly separable dichotomies of the vertices of an n-dimensional cube, consequently bounding the number of decision functions that are representable by naive Bayes classifiers with binary predictors. Domingos and Pazzani (1997) studied the optimality of naive Bayes at length and pointed out that, even if the independence assumption among predictors is violated, naive Bayes could achieve optimality under 0-1 loss. Jaeger (2003) showed, for binary predictors that, classifier expressivity at different levels of complexity is characterized by separability with polynomials of different degrees. Ling and Zhang (2002) reported negative results for the expressive power of Bayesian networks; they proved that a Bayesian network where each node has at most k parents cannot represent any function containing $(k+1)$-XORs. Nakamura et al. (2005)

studied the inner product space for Bayesian network classifiers with binary predictors, that is, the smallest Euclidean space that represents the induced concept class. They obtained upper and lower bounds on the dimension of the inner product space and they linked the dimension of the inner product space with the Vapnik-Chervonekis (VC) dimension (Vapnik and Chervonenkis, 1971). Yang and Wu (2012) studied the case of Bayesian networks with k-valued nodes. They computed the VC dimension for fully connected Bayesian networks and for Bayesian networks without V-structures. In both cases they showed that the VC dimension is equal to the dimension of the inner product space.

In this paper we try to generalize the above results within a unified framework. To do this we compute polynomial threshold functions for Bayesian network (BN) binary classifiers in order to express their decision boundaries. This research is restricted to BN classifiers where the binary class variable, C, has no parents and where the predictors are categorical. As usual, our results extend to non-binary classifiers considering an ensemble of binary classifiers. Polynomial threshold functions are a way to describe the decision boundary of a discrete classifier and are a generalization of the results of Minsky (1961) and Peot (1996). In absence of V-structures in the BN we prove that the obtained families of polynomial representing the induced decision functions form linear spaces that are representations of the inner product spaces. We are able to compute the dimensions of those linear spaces and thus of the inner product space extending the results of Nakamura et al. (2005) and Yang and Wu (2012).

In Section 2 we define the notation used and briefly describe Bayesian network classifiers. In Section 3 we define a polynomial representation of the Iverson bracket (Iverson, 1962) over a finite number of categorical variables and derive the representation of discrete probability functions and of conditional probability tables. We then investigate polynomial representations of decision functions induced by Bayesian network classifiers. We look at Bayesian network classifiers in ascending order of complexity: naive Bayes classifiers in Section 3.2, tree augmented naive Bayes classifiers in Section 3.3, Bayesian network-augmented naive Bayes classifiers in Section 3.4 and fully connected Bayesian network classifiers in Section 3.5. In Section 4 we analyse the expressive power of BAN classifiers. Finally we present our conclusions and suggest possible future works in Section 5.

2. Preliminaries

We will use bold letters, $\mathbf{x}$ or $\mathbf{k}$, to represent elements of a product space, and letters with a subscript to represent the respective components, for example x_2 indicates the second component of $\mathbf{x}$. The capital letter P always refers to a probability, defined on an appropriate measure space, and capital letters X or X_1, X_2, X_i refer to random variables. For every function $f : \Omega \to \mathbb{R}$ and $\Omega_0 \subseteq \Omega$, we write $f_{|\Omega_0}$ for the restriction of f over Ω_0, that is, the function $f_{|\Omega_0} : \Omega_0 \to \mathbb{R}$ such that $f_{|\Omega_0}(\xi) = f(\xi)$ for every $\xi \in \Omega_0$.

We consider a binary classification, that is, we are given a training set of labelled observations $\mathcal{T} = \{(\mathbf{x}^1, c^1), \ldots, (\mathbf{x}^N, c^N)\}$, where $\mathbf{x}^i = (x_1^i, x_2^i, \ldots, x_n^i) \in \Omega \subset \mathbb{R}^n$, with $|\Omega| < \infty$, and classes $c^i \in \{-1, +1\}$. We search for a classification algorithm (classifier) Φ that, once trained on the set $\mathcal{T}$, is able to classify every new instance $\mathbf{x} \in \Omega$ into one of the two classes -1 or $+1$. Every classifier induces a decision function $f_{\mathcal{T}}^{\Phi} : \Omega \to \{-1, +1\}$, where the clas-

sifier Φ will classify each new instance $\mathbf{x}$ to class a if $f_{\mathcal{T}}^{\Phi}(\mathbf{x}) = a$. We drop the subscript $\mathcal{T}$ since we are not interested in the relationship to the training set.

In this paper we focus on Bayes classifiers, probabilistic classifiers which learn from the training set $\mathcal{T}$ a joint probability $P(\mathbf{X}, C)$ and classify each new instance $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ in the most probable a posteriori class (MAP), that is,

$$f^{\Phi}(\mathbf{x}) = arg\max_{c} P(C = c|\mathbf{X} = \mathbf{x}) = arg\max_{c} P(\mathbf{X} = \mathbf{x}, C = c).$$

BN classifiers (Bielza and Larrañaga, 2014) are Bayesian classifiers that factorize the joint probability distribution according to a Bayesian network. They range from the simplest naive Bayes classifier (Figure 1), where the predictor variables are assumed to be conditionally independent given the class variable, to the unrestricted Bayesian classifier, where a general form of Bayesian network (Pearl, 1988) is permitted. We will study only Bayesian network augmented naive Bayes classifiers, that is, we will consider the class C as a root node parent of every predictor variable. Once the structure of the Bayesian network is fixed, we need to estimate the parameters of the probability distribution. Thanks to the factorization implied by the Bayesian network structure we just estimate the conditional probability distributions of every variable given its parents, that is we have to estimate $P(X_i = x_i|\mathbf{X}_{\mathbf{pa}(i)} = \mathbf{x}_{\mathbf{pa}(i)})$, where $\mathbf{X}_{\mathbf{pa}(i)}$ stands for the vector of the parents of X_i. In the discrete case this is reduced to the estimation of conditional probability tables. They could be estimated in several ways, but the straightforward approach using the maximum likelihood estimators (MLE), which are the relative frequencies, could lead to some conditional probabilities equal to zero. A Bayesian approach, such as the Laplace estimator or more generally Dirichlet-prior estimation of the parameters, will avoid this drawback. Because of this observation we will assume from now on that all parameters learned will be different from zero, that is, all the probabilities are positive.

To describe the complexity of decision functions we use the concept of threshold functions.

Definition 1 *Given a decision function $f : \Omega \to \{-1, +1\}$, where $\Omega \subset \mathbb{R}^n$, $|\Omega| < \infty$ and $r : \mathbb{R}^n \mapsto \mathbb{R}$ a polynomial we say that r sign-represents f or that f is computed by a polynomial threshold function, if*

$$f(\mathbf{x}) = sgn(r(\mathbf{x})) \text{ for every } \mathbf{x} \in \Omega.$$

Moreover, given a set of polynomials $\mathcal{P}$, we denote by $sgn(\mathcal{P})$ the set of decision functions that are sign-representable by polynomials in $\mathcal{P}$ and by $\{-1, +1\}^{\Omega}$ the set of all the $2^{|\Omega|}$ decision functions over Ω. Polynomial threshold functions are mainly studied in the theory of Boolean functions, functions $g : \{-1, +1\}^n \to \{-1, +1\}$ (O'Donnell and Servedio, 2010; Wang and Williams, 1991). A particular case is the linear threshold function, that is, when the degree of the polynomial that sign-represents the decision function is equal to one. Observe that different polynomials can sign-represent the same decision function, and not every polynomial sign-represents a decision function. In general we have that a polynomial $r(\mathbf{x})$ sign-represents a decision function over Ω if and only if $r(\mathbf{x}) \neq 0$ for every $\mathbf{x} \in \Omega$.

Example 1 *Consider* $\Omega = \Omega_1 \times \Omega_2$*, with* $\Omega_1 = \{0, 2, 4\}$ *and* $\Omega_2 = \{0, 1\}$*, and a decision function* $f : \Omega \to \{-1, +1\}$ *such that*

$$f(x_1, x_2) = \begin{cases} -1 & \text{if } (x_1, x_2) \in \{(0,0), (2,0), (4,1)\} \\ +1 & \text{if } (x_1, x_2) \in \{(0,1), (2,1), (4,0)\}. \end{cases}$$

If we define polynomials

$$r(x_1, x_2) = -2x_1x_2 + x_1 + 6x_2 - 3$$
$$q(x_1, x_2) = -2x_1^2x_2 + x_1^2 + 16x_2 - 8,$$

we have $sgn(r(x_1, x_2)) = sgn(q(x_1, x_2)) = f(x_1, x_2)$ *for every* $(x_1, x_2) \in \Omega$*, with* $r \neq q$*, thus both polynomials sign-represent* f*.*

If we consider a polynomial $s(x_1, x_2) = x_1^3 + x_2 - 8$*, we have that* $s(2, 0) = 0$ *and thus* $s(x_1, x_2)$ *cannot sign-represent any decision function over* Ω*.*

3. Polynomial Threshold Functions for Bayesian Network Classifiers

We develop a method to easily compute polynomial threshold functions for Bayesian network classifiers. This method is an extension of the well-known results on the decision boundary of naive Bayes classifiers (Minsky, 1961; Peot, 1996). The method is based on the polynomial interpolation of discrete probability functions or equivalently their logarithms. Pistone et al. (2001) give a more formal and general description of this subject, also addressing applications to Bayesian networks. We will develop this method directly using Lagrange basis polynomials.

3.1 Lagrange Interpolation of Discrete Probability

The proofs of the results on the decision boundary in naive Bayes classifiers are based on a representation of the categorical distribution over two values $\{0, 1\}$ in an exponential form, $P(X = x) = p^x(1 - p)^{1-x}$, with $x \in \{0, 1\}$ and $p \in (0, 1)$. We aim to reproduce the same representation for a categorical variable $X \in \Lambda = \{\xi^1, \xi^2, \ldots, \xi^m\} \subset \mathbb{R}$, where the values of variable X are indicated as ξ^j with j as upper index. We consider $\{p(1), \ldots, p(m)\}$ such that $\sum_{j=1}^m p(j) = 1$ and, using the Iverson bracket (Iverson, 1962), we write

$$P(X = x) = \prod_{j=1}^{m} p(j)^{[x=\xi^j]}. \tag{1}$$

If $X \in \{0, 1\}$ we could represent $[x = 0]$ as $1-x$ and $[x = 1]$ as x. If we consider a categorical variable, $X \in \Lambda = \{\xi^1, \xi^2, \ldots, \xi^m\} \subset \mathbb{R}$, we need to find m polynomials $\left\{\ell_j^\Lambda\right\}_{j=1}^m$ such that

$$\ell_j^\Lambda(\xi^j) = 1,$$

and

$$\ell_j^\Lambda(\xi^k) = 0 \text{ for every } k \neq j.$$

We easily see that such polynomials exist and have the following form:

$$\ell_j^{\Lambda}(x) = \prod_{k \neq j} \frac{(x - \xi^k)}{(\xi^j - \xi^k)}. \tag{2}$$

The polynomials defined in Equation (2) are the Lagrange basis polynomials (Abramowitz and Stegun, 1964; Jeffreys and Jeffreys, 1999) over the points in Λ. These polynomials are m linearly independent polynomials of degree $m-1$, and so they form a basis of polynomials in one variable whose degree is at most $m-1$. We summarize some properties of these polynomials in the following lemma.

Lemma 2 *Let $\Omega_i = \{\xi_i^1, \xi_i^2, \ldots, \xi_i^{m_i}\} \subset \mathbb{R}$, for $i = 1, \ldots, n$. For every i define the Lagrange basis, $\left\{\ell_j^{\Omega_i}(x_i)\right\}$, over Ω_i as in Equation (2). Then we have*

1. *For every $i = 1, \ldots, n$, $\left\{\ell_j^{\Omega_i}(x_i)\right\}_{j=1}^{m_i}$ form a basis of the space of polynomials in x_i of degree $|\Omega_i| - 1$.*
2. *$\sum_{j_{i_1}=1}^{m_{i_1}} \sum_{j_{i_2}=1}^{m_{i_2}} \cdots \sum_{j_{i_l}=1}^{m_{i_l}} \prod_{s \in I} \ell_{j_s}^{\Omega_s}(x_s) = \prod_{i \in I} \sum_{j_i=1}^{m_i} \ell_{j_i}^{\Omega_i}(x_i) = 1$, for every $\mathbf{x} \in \mathbb{R}^I$ and for all $I = \{i_1, \ldots, i_l\} \subseteq \{1, \ldots, n\}$.*
3. *$\prod_{i \in I} \ell_{j_i}^{\Omega_i}(x_i) = [x_i = \xi_i^{j_i} \ \forall i \in I]$, for every $I \subseteq \{1, \ldots, n\}$, for all $\{j_i\}_{i \in I}$ such that $1 \leq j_i \leq m_i$, and for every $\mathbf{x} \in \times_{i \in I} \Omega_i$.*
4. *$\sum_{j_{i_1}=1}^{m_{i_1}} \sum_{j_{i_2}=1}^{m_{i_2}} \cdots \sum_{j_{i_p}=1}^{m_{i_p}} \prod_{s \in I} \ell_{j_s}^{\Omega_s}(x_s) = \prod_{i \in I \setminus J} \ell_{j_i}^{\Omega_i}(x_i)$, for every $\mathbf{x} \in \mathbb{R}^I$ and for all $J = \{i_i, \ldots, i_p\} \subset I \subseteq \{1, \ldots, n\}$.*

Proof The proof of the above lemma is trivial, and we just outline some points. Point 1 follows from the linear independences of the Lagrange basis polynomials. To prove point 2, we have merely to observe that, since $\left\{\ell_j^{\Omega_i}\right\}_{j=1}^{m_i}$ is a basis, we have that the polynomial constant 1 admits a unique representation in the considered basis, in particular $1 = \sum_{j=1}^{m_i} \ell_j^{\Omega_i}(x_i)$. Point 3 follows trivially by substitution. To prove point 4 we apply point 2 as follows,

$$\sum_{j_{i_1}=1}^{m_{i_1}} \sum_{j_{i_2}=1}^{m_{i_2}} \cdots \sum_{j_{i_p}=1}^{m_{i_p}} \prod_{s \in I} \ell_{j_s}^{\Omega_s}(x_s) = \underbrace{\left(\sum_{j_{i_1}=1}^{m_{i_1}} \sum_{j_{i_2}=1}^{m_{i_2}} \cdots \sum_{j_{i_p}=1}^{m_{i_p}} \prod_{s \in J} \ell_{j_s}^{\Omega_s}(x_s) \right)}_{=1} \prod_{i \in I \setminus J} \ell_{j_i}^{\Omega_i}(x_i) = \prod_{i \in I \setminus J} \ell_{j_i}^{\Omega_i}(x_i).$$

■

If we are given a categorical random variable X over $\Lambda = \{\xi^1, \ldots, \xi^m\}$ whose probability mass function is P, we are able to rewrite Equation (1) using the Lagrange basis, as

$$P(X = x) = \prod_{j=1}^{m} p(j)^{[x=\xi^j]} = \prod_{j=1}^{m} p(j)^{\ell_j^{\Lambda}(x)}, \tag{3}$$

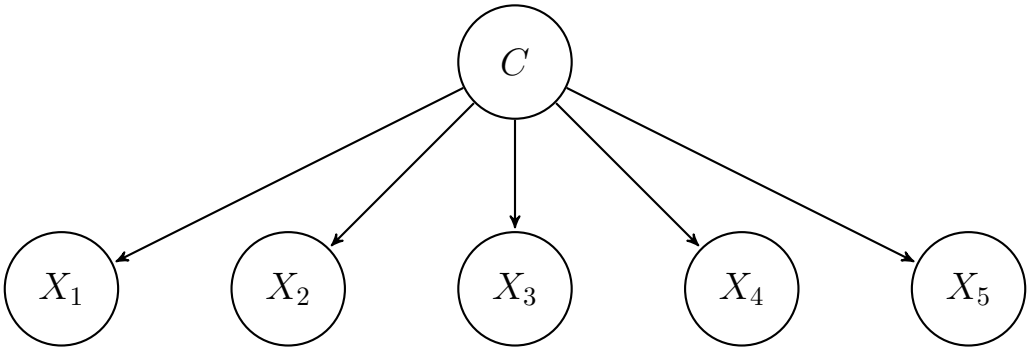


Figure 1: Naive Bayes classifier structure with five predictor variables

where $p(j) = P(X = \xi^j)$ are the values of the probability mass function over Λ. Equation (3) is a consequence of the identity $[x = \xi^j] = \ell_j^\Lambda(x)$ which derives from point 3 of Lemma 2 considering $|I| = 1$. More generally, we consider a set of random variables $\{X_1, X_2, \ldots, X_n\}$ such that, for every $i = 1, \ldots, n$, the variable $X_i \in \Omega_i = \{\xi_i^1, \xi_i^2, \ldots, \xi_i^{m_i}\}$. If we are given a conditional probability table that represents the probability function $P(X_1 = x_1 | X_2 = x_2, \ldots, X_n = x_n)$, we can use the Iverson bracket over n variables $x_1, \ldots, x_n$ to describe the conditional distribution of X_1 given $X_2, \ldots, X_n$,

$$P\left(X_1 = x_1 | X_2 = x_2, \ldots, X_n = x_n\right) = \prod_{(j_1, \ldots, j_n)} p(j_1 | j_2, \ldots, j_n)^{[x_i = \xi_i^{j_i} \ \forall i = 1, \ldots, n]},$$

where $p(j_1|j_2, \ldots, j_n) = P(X_1 = \xi_1^{j_1} | X_2 = \xi_2^{j_2}, \ldots, X_n = \xi_n^{j_n})$ are the values of the conditional probability table. Now using point 3 of Lemma 2 with $I = \{1, \ldots, n\}$, we get

$$P\left(X_1 = x_1 | X_2 = x_2, \ldots, X_n = x_n\right) = \prod_{(j_1, \ldots, j_n)} p(j_1 | j_2, \ldots, j_n)^{\prod_{i=1}^m \ell_{j_i}^{\Omega_i}(x_i)}. \tag{4}$$

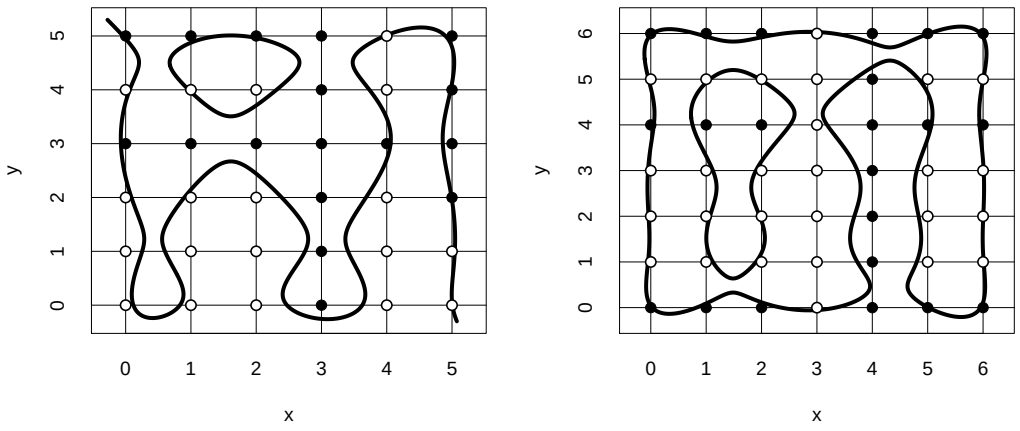
3.2 Naive Bayes

We consider a naive Bayes classifier (NB) (Figure 1) where the predictor variables $X_i \in \Omega_i$ are conditionally independent given the class variable C. The joint probability distribution factorizes as follows:

$$P(C = c, X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n) = P(C = c) \prod_{i=1}^n P(X_i = x_i | C = c). \tag{5}$$

If the predictor variables are binary, Minsky (1961) proved that the decision boundaries are hyperplanes. For categorical predictors, the scenario is much more complicated as shown in Figure 2.

Theorem 3 *A decision function f for a binary classification problem over n categorical variables $X_i \in \Omega_i = \{\xi_i^1, \ldots, \xi_i^{m_i}\}$, with $|\Omega_i| = m_i$, is sign-represented by a polynomial of the form $\sum_{i=1}^n \left(\sum_{j=1}^{m_i} \alpha_i(j) \ell_j^{\Omega_i}(x_i) \right)$ if and only if there exists a naive Bayes classifier, with probability tables without zeros entries, that induces f, where $\ell_j^{\Omega_i}$ are the Lagrange basis over Ω_i.*



(a) $X, Y \in \{0, 1, \ldots, 5\}$ (b) $X, Y \in \{0, 1, \ldots, 6\}$

Figure 2: Decision boundary for two example, (a) and (b), of naive Bayes classifiers with two categorical variables X, Y. Boundaries are computed as location of zeroes of polynomials built as in Theorem 3

Proof We consider a naive Bayes classifier as in Figure 1. For every $i = 1, \ldots, n$ the variable X_i takes values over $\Omega_i = \{\xi_i^1, \ldots, \xi_i^{m_i}\}$, a subset of $\mathbb{R}$ of cardinality m_i. Thanks to Equation (3), we can express, for every value c of the class, the conditional probability $P(X_i|C)$ as

$$P(X_i = x_i|C = c) = \prod_{j=1}^{m_i} p_i(j|c)^{\ell_j^{\Omega_i}(x_i)},$$

where $p_i(j|c) = P(X_i = \xi_i^j|C = c)$. If we define $a_i(j|c) = \ln(p_i(j|c))$, and assuming that $p_i(j|c) > 0$, we have that

$$P(X_i = x_i|C = c) = \exp\left(\sum_{j=1}^{m_i} a_i(j|c)\ell_j^{\Omega_i}(x_i)\right). \tag{6}$$

Using this representation we easily find the decision function for NB with arbitrary discrete predictor variables. Setting $a = \ln(P(C = +1))$ and $b = \ln(P(C = -1))$, we have that a new instance $\mathbf{x} = (x_1, \ldots, x_n)$ will be classified as $C = +1$ if

$$P(X_1 = x_1, \ldots, X_n = x_n, C = +1) > P(X_1 = x_1, \ldots, X_n = x_n, C = -1).$$

Using Equations (5) and (6) we have that the previous inequality could be rewritten as

$$\exp\left(a + \sum_{i=1}^{n}\left(\sum_{j=1}^{m_i} a_i(j|+1)\ell_j^{\Omega_i}(x_i)\right)\right) > \exp\left(b + \sum_{i=1}^{n}\left(\sum_{j=1}^{m_i} a_i(j|-1)\ell_j^{\Omega_i}(x_i)\right)\right),$$

so the decision function for a naive Bayes classifier is

$$f^{NB}(\mathbf{x}) = sgn\left(a - b + \sum_{i=1}^{n}\left(\sum_{j=1}^{m_i}\alpha'_i(j)\ell_j^{\Omega_i}(x_i)\right)\right), \tag{7}$$

where $\alpha'_i(j) = a_i(j|+1) - a_i(j|-1) = \ln\left(\frac{P(X_i=\xi_i^j|C=+1)}{P(X_i=\xi_i^j|C=-1)}\right)$. We see from Equation (7) that the decision function is sign-represented by a polynomial that admits the representation $\sum_{i=1}^{n}\left(\sum_{j=1}^{m_i}\alpha_i(j)\ell_j^{\Omega_i}(x_i)\right)$. In fact we have that the $a - b = \ln\left(\frac{P(C=+1)}{P(C=-1)}\right)$ term could be included in the summation using Lemma 2, for example with the following choice of coefficient,

$$\alpha_i(j) = \ln\left(\frac{P(X_i = \xi_i^j|C=+1)}{P(X_i=\xi_i^j|C=-1)}\right) + k_i \ln\left(\frac{P(C=+1)}{P(C=-1)}\right), \tag{8}$$

where $\sum_{i=1}^{n} k_i = 1$. We have proved the *if* part of the theorem.

To prove the *only if* we have just to observe that choosing the conditional probabilities for the predictor variables given the class, $P(X_i = \xi_i^j|C=c)$, the probability mass for the class $P(C=+1) = 1 - P(C=-1)$, and the values of $\{k_i\}_{i=1}^n$ we are able to adjust the coefficients $\alpha_i(j)$ in (8) to any possible values in $\mathbb{R}$. For example the following choices are sufficient

$$\begin{aligned}
&P(X_i = \xi_i^j|C=-1) = \frac{1}{m_i} \quad \forall i = 1,\dots,n \text{ and } j = 1,\dots,m_i,\\
&P(X_i = \xi_i^j|C=+1) = \frac{e^{\alpha_i(j)}}{\sum_{j=1}^{m_i} e^{\alpha_i(j)}} \quad \forall i = 1,\dots,n \text{ and } j = 1,\dots,m_i,\\
&k_i = \frac{\ln\left(\frac{1}{m_i}\sum_{j=1}^{m_i} e^{\alpha_i(j)}\right)}{\sum_{i=1}^{n}\ln\left(\frac{1}{m_i}\sum_{j=1}^{m_i} e^{\alpha_i(j)}\right)} \quad \forall i = 1,\dots,n,\\
&\ln\left(\frac{P(C=+1)}{P(C=-1)}\right) = \sum_{i=1}^{n}\ln\left(\frac{1}{m_i}\sum_{j=1}^{m_i} e^{\alpha_i(j)}\right).
\end{aligned}$$

■

As a result of Theorem 3 we have that a naive Bayes classifier could represent every decision function which is sign-representable by a polynomial of the family

$$\left\{r(\mathbf{x}) = \sum_{i=1}^{n}\left(\sum_{j=1}^{m_i}\alpha_i(j)\ell_j^{\Omega_i}(x_i)\right),\ \alpha_i(j)\in\mathbb{R}\right\}.$$

Only if we fix the prior probability over the class C are there restrictions on the coefficients $\alpha_i(j)$.

Corollary 4 *Let f be a decision function for a binary classification problem with n categorical predictor variables $X_i \in \Omega_i = \{\xi_i^1,\dots,\xi_i^{m_i}\} \subset \mathbb{R}$. The following sentences are equivalent:*

X_1	$C=-1$	$C=+1$
0	0.3	0.3
1	0.1	0.2
2	0.4	0.1
3	0.1	0.2
4	0.1	0.2

X_2	$C=-1$	$C=+1$
0	0.2	0.4
1	0.1	0.2
2	0.7	0.4

Table 1: Conditional Probability Tables in Example 2

i) *f is sign-represented by a polynomial of the form $\sum_{i=1}^{n}\left(\sum_{j=1}^{m_i}\alpha_i(j)\ell_j^{\Omega_i}(x_i)\right)$ with $\alpha_i(j)$ such that for every $i=1,\ldots,n$, there exists $j_{i,1}$ and $j_{i,2}$ such that $\alpha_i(j_{i,1})<0$ and $\alpha_i(j_{i,2})>0$ or alternatively $e^{\alpha_i(j)}=1$ for every $j=1,\ldots,m_i$.*

ii) *There exists a naive Bayes classifier, with probability tables without zeros entries, that induces f, with uniform prior probability over the class C.*

Proof The corollary follows from (8) in proof of Theorem 3, it is easy to show that the two conditions are equivalent. ■

As we can see, the coefficients $\alpha_i(j)$ are related to the probability model underlying the problem, and are usually estimated from the training set but they do not generally assure the minimization of classification errors. An interesting model to deal with this problem is the weighted naive Bayes classifier (Webb and Pazzani, 1998; Hall, 2007). Weights are introduced in the probability factorization,

$$P(C=c|\mathbf{X}=\mathbf{x}) \propto w_c P(C=c)\prod_{i=1}^{n}\left[P(X_i=x_i|C=c)\right]^{w_i},$$

and thus the decision function has the same form as in (7), but with modified coefficients

$$\alpha_i(j)=w_i\ln\frac{P(X_i=j|C=+1)}{P(X_i=j|C=-1)}.$$

Note that introducing the weights in the model does not change the form of the polynomial sign-representing the decision functions, so it does not improve the expressive power of the model. Even so, using the weighted model it is possible to search for polynomials that minimize the misclassification and improve accuracy (Zaidi et al., 2013). As future research it may be of some interest to study how to search polynomials to directly minimize the misclassification error and how this reflects on the implicitly defined NB classifier.

Example 2 *We consider a naive Bayes classifier with two predictor variables $X_1\in\Omega_1=\{0,1,2,3,4\}$ and $X_2\in\Omega_2=\{0,1,2\}$. We have a uniform prior probability over the class C, that is, $P(C=-1)=P(C=+1)=0.5$, and we consider the conditional probability tables for X_1 and X_2 given in Table 1. We can directly build the polynomial threshold functions $r(x_1,x_2)$ that sign-represent the decision function induced by this classifier. The related*

$\alpha_1(0) = \ln \frac{0.3}{0.3} = 0$	$\alpha_2(0) = \ln \frac{0.4}{0.2} = \ln 2$
$\alpha_1(1) = \ln \frac{0.2}{0.1} = \ln 2$	$\alpha_2(1) = \ln \frac{0.2}{0.1} = \ln 2$
$\alpha_1(2) = \ln \frac{0.1}{0.4} = -\ln 4$	$\alpha_2(2) = \ln \frac{0.4}{0.7} = -\ln \frac{7}{4}$
$\alpha_1(3) = \ln \frac{0.2}{0.1} = \ln 2$	
$\alpha_1(4) = \ln \frac{0.2}{0.1} = \ln 2$	

Table 2: Coefficients computations of polynomial (9)

coefficients are $\alpha_1(j) = \ln \frac{P(X_1=j|C=+1)}{P(X_1=j|C=-1)}$ *and* $\alpha_2(j) = \ln \frac{P(X_2=j|C=+1)}{P(X_2=j|C=-1)}$*, and the polynomial* $r(x_1, x_2)$ *is*

$$r(x_1, x_2) = \sum_{j=0}^{4} \alpha_1(j) \ell_j^{\Omega_1}(x_1) + \sum_{j=0}^{2} \alpha_2(j) \ell_j^{\Omega_2}(x_2). \tag{9}$$

The computations of the coefficients are shown in Table 2. We have that the polynomial threshold function in Equation (9), expressed with the Lagrange basis, is

$$\begin{aligned} r(x_1, x_2) =& \frac{x_1(x_1-2)(x_1-3)(x_1-4)}{-6} \ln 2 - \frac{x_1(x_1-1)(x_1-3)(x_1-4)}{4} \ln 4 \\ &+ \frac{x_1(x_1-1)(x_1-2)(x_1-4)}{-6} \ln 2 + \frac{x_1(x_1-1)(x_1-2)(x_1-3)}{24} \ln 2 \\ &+ \frac{(x_2-1)(x_2-2)}{2} \ln 2 + \frac{x_2(x_2-2)}{-1} \ln 2 - \frac{x_2(x_2-1)}{2} \ln \frac{7}{4}. \end{aligned}$$

We observe that the above polynomial satisfies the condition of Corollary 4, as it should because the prior probability over C *is uniform. Figure 3 shows the decision boundary induced by* $r(x_1, x_2)$*.*

3.3 Tree Augmented Naive Bayes

We now consider a tree augmented naive Bayes (TAN) classifier (Friedman et al., 1997) as shown in Figure 4. In this model, a predictor variable $X_i \in \Omega_i = \{\xi_i^1, \ldots, \xi_i^{m_i}\}$ is allowed to have at most two parents, the class C and an other variable, $X_{pa(i)} \in \Omega_{pa(i)}$. The joint probability distribution of $(C, X_1, X_2, \ldots, X_n)$ over $\{-1, +1\} \times \Omega_1 \times \cdots \times \Omega_n$ can be factorized according to the Bayesian network theory as

$$P(C = c) \prod_{i=1}^{n} P\left(X_i = x_i | C = c, X_{pa(i)} = x_{pa(i)}\right). \tag{10}$$

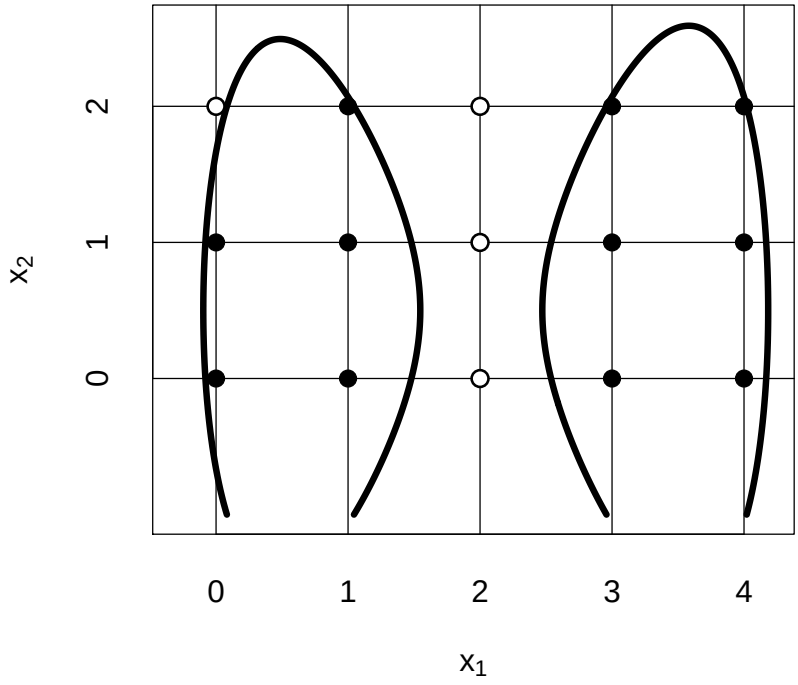


Figure 3: Decision boundary for the naive Bayes structure of Example 2

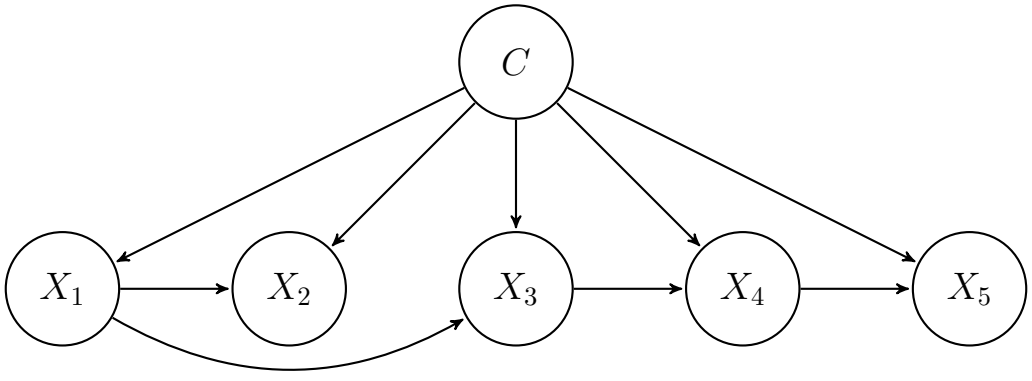


Figure 4: Tree augmented naive Bayes classifier structure with five predictor variables

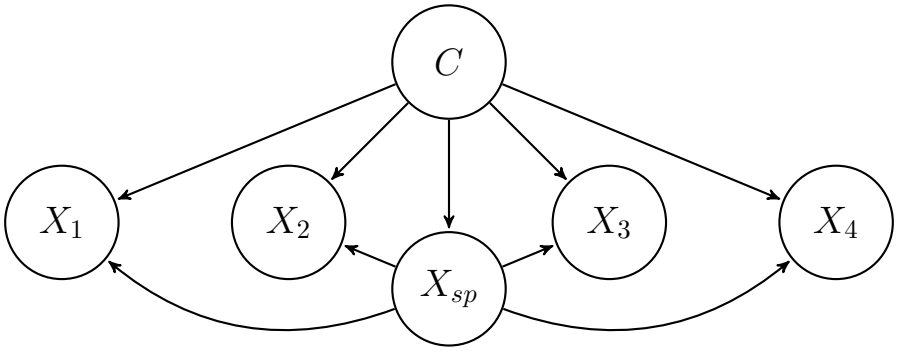


Figure 5: SPODE Bayes classifier structure with five predictor variables

We can write down a similar representation to the NB case. For each $i = 1, \ldots, n$, we apply Equation (4) and obtain

$$P\left(X_i = x_i | C = c, X_{pa(i)} = x_{pa(i)}\right) = \prod_{j=1}^{m_i} \prod_{k=1}^{m_{pa(i)}} p_i(j|c,k)^{\left(\ell_k^{\Omega_{pa(i)}}(x_{pa(i)})\ell_j^{\Omega_i}(x_i)\right)}. \tag{11}$$

We can now prove, combining Equations (10) and (11), a result similar to the NB case.

Lemma 5 *If f^{TAN} is the decision function induced by a TAN for a binary classification problem with n categorical predictor variables $\{X_i \in \Omega_i\}_{i=1}^n$ and with probability tables without zeros entries, then there exists a polynomial, of the form*

$$\sum_{i=1}^{n} \sum_{j=1}^{m_i} \ell_j^{\Omega_i}(x_i) \sum_{k=1}^{m_{pa(i)}} \beta_i(j|k) \ell_k^{\Omega_{pa(i)}}(x_{pa(i)}),$$

that sign-represents f^{TAN}, where we consider $\sum_{k=1}^{m_{pa(i)}} \beta_i(j|k)\ell_k^{\Omega_{pa(i)}}(x_{pa(i)}) = \beta_i(j)$ when $\Omega_{pa(i)} = \emptyset$, that is, when class C is the only parent of a node (the root node of the tree).

Proof The proof is a straightforward computation of the logarithm of Equation (10) using Equation (11) and the definition $\beta_i(j|k) = \ln\left(\frac{p_i(j|+1,k)}{p_i(j|-1,k)}\right)$. The term corresponding to the probability over the class $\ln\left(\frac{P(C=+1)}{P(C=-1)}\right)$ could be made vanishing into the coefficients of the root node X_t of the tree, using point 2 of Lemma 2 with $I = \{t\}$, with the following choice of coefficients

$$\beta_t(j) = \ln\left(\frac{p_i(j|+1)}{p_i(j|-1)}\right) + \ln\left(\frac{P(C=+1)}{P(C=-1)}\right).$$

■

A particular case of TAN is the *SuperParent-One-Dependence Estimator* (SPODE) (Keogh and Pazzani, 2002), where all the predictors depend on the same predictor (superparent) (Figure 5). The joint distribution factorizes as follows:

$$P\left(C = c\right) P\left(X_{sp} = x_{sp} | C = c\right) \prod_{i \neq sp} P\left(X_i = x_i | C = c, X_{sp} = x_{sp}\right),$$

where X_{sp} stands for the superparent node. In this case, the representation of Lemma 5 reduces to

$$f^{SPODE}(\mathbf{x}) = sgn\left(\sum_{i\neq sp}\sum_{j=1}^{m_i}\ell_j^{\Omega_i}(x_i)\sum_{k=1}^{m_{sp}}\beta_i(j|k)\ell_k^{\Omega_{sp}}(x_{sp})\right), \tag{12}$$

where f^{SPODE} is the induced decision function. If we fix the superparent node, we have a stronger characterization of the induced decision functions, the analogue of Theorem 3.

Theorem 6 *A decision function for a binary classification problem over categorical predictor variables is sign-represented by a polynomial of the form*

$$\sum_{i\neq sp}\sum_{j=1}^{m_i}\ell_j^{\Omega_i}(x_i)\sum_{k=1}^{m_{sp}}\beta_i(j|k)\ell_k^{\Omega_{sp}}(x_{sp}),$$

if and only if it is induced by a SPODE classifier with X_{sp} as the superparent node and with probability tables without zeros entries.

Proof The *if* part of the theorem is precisely Equation (12). To prove the *only if* part we repeat a similar argument as in Theorem 3. We observe (Lemma 2, point 4, with $J=\{i\}$ and $I=\{i,sp\}$) that for every $i\neq sp$,

$$\ell_k^{\Omega_{sp}}(x_{sp}) = \sum_{j=1}^{m_i}\ell_j^{\Omega_i}(x_i)\ell_k^{\Omega_{sp}}(x_{sp}),$$

and so the coefficient $\beta_i(j|k)$ could be seen as

$$\beta_i(j|k) = \ln\left(\frac{P(X_i=j|X_{sp}=k,C=+1)}{P(X_i=j|X_{sp}=k,C=-1)}\right)+\alpha_i(k),$$

where $\sum_{i\neq sp}\alpha_i(k) = \ln\left(\frac{P(X_{sp}=\xi_{sp}^k|C=+1}{P(X_{sp}=\xi_{sp}^k|C=-1}\right)+\alpha$ and $\alpha=\ln\left(\frac{P(C=+1)}{P(C=-1)}\right)$. Then adjusting $\alpha_i(k)$ and α properly we can find a SPODE model, that is, probability distributions over the predictors and the class that induces

$$f = sgn\left(\sum_{i\neq sp}\sum_{j=1}^{m_i}\ell_j^{\Omega_i}(x_i)\sum_{k=1}^{m_{sp}}\beta_i(j|k)\ell_k^{\Omega_{sp}}(x_{sp})\right),$$

for every $\beta_i(j|k)\in\mathbb{R}$. ■

Remark 7 *We observe that, as for Theorem 3, the proof of Theorem 6 adds free parameters to the model. For every variable we modify the related coefficients and then we adjust the modifications with the parent coefficients. As in the proof of Theorem 3 we are able to use the added parameters to define proper probability distributions, that is to make the defined probability add up to one.*

Remark 8 *Results similar to Theorem 6 could be proved whenever the structure of the predictor sub-graph of a TAN classifier is fixed. We expound no further theorems about TAN classifiers, as, in the next section, we will prove a more general result, of which NB and TAN are special cases.*

Example 3 *We look at the SPODE model (see Figure 6 for structure) with the superparent node X_{sp}. We consider $X_1 \in \{0,1,2\}$, $X_2 \in \{0,1,2,3\}$ and $X_{sp} \in \{0,1\}$ with conditional probability tables as shown in Table 3. The polynomial threshold function $r(x_{sp}, x_1, x_2)$ can be computed directly as specified in Lemma 5:*

$$
\begin{aligned}
r(x_sp, x_1, x_2) &= (1-x_{sp}) \ln\left(\frac{0.4}{0.8}\right) + x_{sp} \ln\left(\frac{0.6}{0.2}\right) \\
&+ (1-x_{sp}) \left(\frac{(1-x_1)(2-x_1)}{2} \ln\left(\frac{0.2}{0.1}\right) + x_1(2-x_1) \ln\left(\frac{0.7}{0.1}\right) + \frac{x_1(x_1-1)}{2} \ln\left(\frac{0.1}{0.8}\right) \right) \\
&+ x_{sp} \left(\frac{(1-x_1)(2-x_1)}{2} \ln\left(\frac{0.7}{0.3}\right) + x_1(2-x_1) \ln\left(\frac{0.1}{0.2}\right) + \frac{x_1(x_1-1)}{2} \ln\left(\frac{0.2}{0.5}\right) \right) \\
&+ (1-x_{sp}) \left(\frac{x_2(2-x_2)(3-x_2)}{2} \ln\left(\frac{0.3}{0.2}\right) + \frac{x_2(x_2-1)(x_2-2)}{6} \ln\left(\frac{0.1}{0.2}\right) \right) \\
&+ x_{sp} \left(\frac{(1-x_2)(2-x_2)(3-x_2)}{6} \ln\left(\frac{0.2}{0.5}\right) + \frac{x_2(x_2-1)(3-x_2)}{2} \ln\left(\frac{0.5}{0.2}\right) \right).
\end{aligned}
$$

We observe that some elements of the Lagrange bases do not appear in $r(x_{sp}, x_1, x_2)$ because the corresponding coefficients are zero, since the conditional probabilities given C are equal.

3.4 Bayesian Network-Augmented Naive Bayes

If the predictor sub-graph can be a generic Bayesian network, we have a Bayesian network-augmented naive Bayes (BAN) classifier. In this case the joint probability distribution is factorized as follows:

$$
P(C=c) \prod_{i=1}^{n} P\left(X_i = x_i | C = c, \mathbf{X}_{\mathbf{pa}(i)} = \mathbf{x}_{\mathbf{pa}(i)}\right), \tag{13}
$$

where $\mathbf{X}_{\mathbf{pa}(i)}$ denotes the vector of the parent variables of X_i that are not C. From now on we will write $\mathbf{pa}(i)$ for the set of indexes defining X_i's parents that are not C and $\mathbb{M}_i = \times_{s \in \mathbf{pa}(i)} \{1, \ldots, m_s\}$ for the set of possible configurations of the parents of X_i. Applying the same arguments as in previous sections we can prove the lemma below.

Lemma 9 *If f^{BAN} is the decision function induced by a BAN classifier for a binary classification problem with n categorical predictors variables $\{X_i \in \Omega_i \subset \mathbb{R}, |\Omega_i| = m_i\}_{i=1}^n$ and with probability tables without zeros entries, then there exists a polynomial of the form*

$$
\sum_{i=1}^{n} \sum_{j=1}^{m_i} \ell_j^{\Omega_i}(x_i) \sum_{\mathbf{k} \in \mathbb{M}_i} \beta_i(j|\mathbf{k}) \prod_{s \in \mathbf{pa}(i)} \ell_{k_s}^{\Omega_s}(x_s),
$$

which sign-represents f^{BAN}, where we write $\sum_{\mathbf{k} \in \mathbb{M}_i} \beta_i(j|\mathbf{k}) \prod_{s \in \mathbf{pa}(i)} \ell_{k_s}^{\Omega_s}(x_s) = \beta_i(j)$ when a variable does not have parents that are not C, that is, $\mathbf{pa}(i) = \emptyset$.

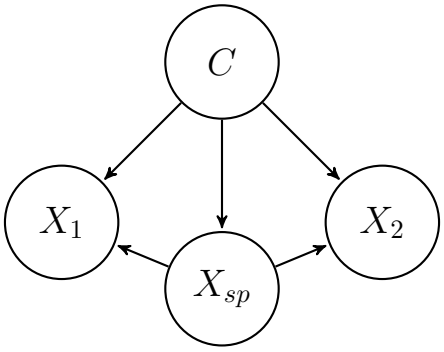


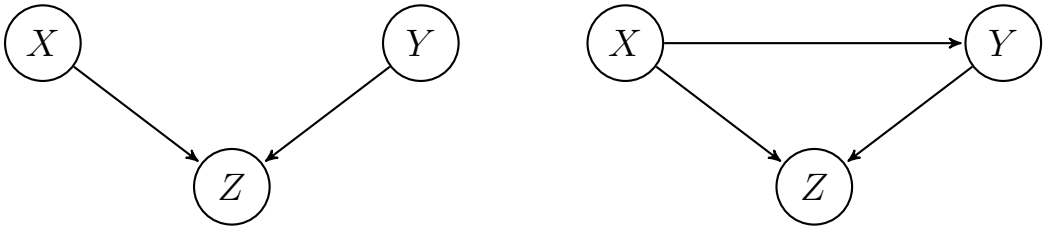
Figure 6: SPODE classifier structure, Example 3

X_{sp}	$C=-1$	$C=+1$
0	0.8	0.4
1	0.2	0.6

X_1	$C=-1$		$C=+1$	
	$X_{sp}=0$	$X_{sp}=1$	$X_{sp}=0$	$X_{sp}=1$
0	0.1	0.3	0.2	0.7
1	0.1	0.2	0.7	0.1
2	0.8	0.5	0.1	0.2

X_2	$C=-1$		$C=+1$	
	$X_{sp}=0$	$X_{sp}=1$	$X_{sp}=0$	$X_{sp}=1$
0	0.5	0.5	0.5	0.2
1	0.2	0.2	0.3	0.2
2	0.1	0.2	0.1	0.5
3	0.2	0.1	0.1	0.1

Table 3: Conditional probability tables in Example 3



(a) (b)

Figure 7: Graphical representation of (a) a V-structure and (b) an example which is not a V-structure

Proof Given a BAN model over predictors $X_i \in \Omega_i = \{\xi_i^1, \ldots, \xi_i^{m_i}\}$, we define

$$\beta_i(j|\mathbf{k}) = \ln\left(\frac{P\left(X_i = \xi_i^j | C = +1, X_s = \xi_s^{k_s}, \forall s \in \mathbf{pa}(i)\right)}{P\left(X_i = \xi_i^j | C = -1, X_s = \xi_s^{k_s}, \forall s \in \mathbf{pa}(i)\right)}\right).$$

Using Equation (4) and taking the logarithm of Equation (13) we obtain the polynomial representation. The additional constant term due to the prior probability over the class, $\ln\left(\frac{P(C=+1)}{P(C=-1)}\right)$, could be embedded into the $\beta_i(j|\mathbf{k})$ coefficients using point 2 of Lemma 2 as in the proofs of Theorem 3 and Lemma 5. ■

Generally speaking, it is not always possible to prove results similar to Theorem 3 or Theorem 6 for BAN classifiers, when decision functions are completely characterized by the set of sign-representing polynomials. Like Yang and Wu (2012), we find that problems arise in the presence of V-structures (Figure 7a) in the predictor sub-graph. A V-structure appears when two nodes share the same child, but are not directly connected. In absence of V-structures we can prove the following result, which extends the previous ones.

Theorem 10 *Let $\mathcal{G}$ be a directed acyclic graph with node X_i for $i = 1, \ldots, n$, and let f be a decision function for a binary classification problem over predictor variables $X_i \in \Omega_i = \{\xi_i^1, \ldots, \xi_i^{m_i}\}$. Suppose that $\mathcal{G}$ does not contain V-structures, then we have that f is sign-represented by the following polynomial*

$$r(\mathbf{x}) = \sum_{i=1}^{n}\sum_{j=1}^{m_i} \ell_j^{\Omega_i}(x_i) \sum_{\mathbf{k}\in\mathbb{M}_i} \beta_i(j|\mathbf{k}) \prod_{s\in\mathbf{pa}(i)} \ell_{k_s}^{\Omega_s}(x_s),$$

if and only if f is induced by a BAN classifier whose predictor sub-graph is $\mathcal{G}$ and with probability tables without zeros entries.

Proof We merely have to prove the *only if* because the *if* implication is precisely Lemma 9. Given a polynomial of the form

$$r(\mathbf{x}) = \sum_{i=1}^{n} \sum_{j\in\Omega_i} \ell_j^{\Omega_i}(x_i) \sum_{\mathbf{k}\in\mathbb{M}_i} \beta_i(j|\mathbf{k}) \prod_{s\in\mathbf{pa}(i)} \ell_{k_s}^{\Omega_s}(x_s),$$

we have to find a BAN classifier inducing $sgn(r(\mathbf{x}))$, whose predictor sub-graph is $\mathcal{G}$. We just have to define the conditional probability distribution of every variable given its parents, since the structure of the BAN is already fixed by $\mathcal{G}$. For every $i = 1, \ldots, n$, we observe that the sub-graph of the parents of X_i is a fully connected Bayesian network, otherwise we will have a V-structure on $\mathcal{G}$. For every i, we can rewrite using point 4 of Lemma 2 the ith addend on the summation,

$$\sum_{j\in\Omega_i} \ell_j^{\Omega_i}(x_i) \sum_{\mathbf{k}\in\mathbb{M}_i} \beta_i(j|\mathbf{k}) \prod_{s\in\mathbf{pa(i)}} \ell_{k_s}^{\Omega_s}(x_s) + \sum_{\mathbf{k}\in\mathbb{M}_i} \alpha_i(\mathbf{k}) \prod_{s\in\mathbf{pa}(i)} \ell_{k_s}^{\Omega_s}(x_s) - \sum_{\mathbf{k}\in\mathbb{M}_i} \alpha_i(\mathbf{k}) \prod_{s\in\mathbf{pa}(i)} \ell_{k_s}^{\Omega_s}(x_s)$$
$$= \sum_{j\in\Omega_i} \ell_j^{\Omega_i}(x_i) \sum_{\mathbf{k}\in\mathbb{M}_i} (\beta_i(j|\mathbf{k}) + \alpha_i(\mathbf{k})) \prod_{s\in\mathbf{pa}(i)} \ell_{k_s}^{\Omega_s}(x_s) - \sum_{\mathbf{k}\in\mathbb{M}_i} \alpha_i(\mathbf{k}) \prod_{s\in\mathbf{pa}(i)} \ell_{k_s}^{\Omega_s}(x_s).$$

Using the *free parameters* $\alpha_i(\mathbf{k})$, it is possible to find for every $\mathbf{k}$, $p_i(j|\mathbf{k}, +1)$ and $p_i(j|\mathbf{k}, -1) \in (0, 1)$ such that

$$\sum_{j=1}^{m_i} p_i(j|\mathbf{k}, +1) = \sum_{j=1}^{m_i} p_i(j|\mathbf{k}, -1) = 1$$
$$\beta_i(j|\mathbf{k}) + \alpha_i(\mathbf{k}) = \ln \frac{p_i(j|\mathbf{k}, +1)}{p_i(j|\mathbf{k}, -1)}.$$

To avoid changing the polynomial $r(\mathbf{x})$, we have to subtract

$$\sum_{\mathbf{k}\in\mathbb{M}_i} \alpha_i(\mathbf{k}) \prod_{s\in\mathbf{pa}(i)} \ell_{k_s}^{\Omega_s}(x_s)$$

from another addend on the summation. Because the parents of X_i are fully connected, we have that among the other addends of $r(\mathbf{x})$, apart from the ith, there is one product that contains $\prod_{s\in\mathbf{pa}(i)} \ell_{k_s}^{\Omega_s}(x_s)$ and so we just subtract $\alpha_i(\mathbf{k})$ from the related coefficient. Iterating the above procedure for all the nodes of the graph $\mathcal{G}$, we are able to build a probability distribution over $X_1, X_2, \ldots, X_n, C$ that satisfies the Bayesian network structure given by $\mathcal{G}$. More precisely, setting

$$P\left(X_i = \xi_i^j | C = c, X_s = \xi_s^{k_s}, \forall s \in \mathbf{pa}(i)\right) = p_i(j|\mathbf{k}, c),$$

we obtain the target BAN model. ■

We observe that the meaning of the representation in Theorem 10 is intuitive. If, as usual, we denote by $\mathbf{pa}(i)$ the function, dependent on $\mathcal{G}$, that maps each variable X_i to the set of its parents, we have that a new instance $\mathbf{x} = (\xi_1^{j_1}, \ldots, \xi_1^{j_n})$ of the predictors will be classified as $C = +1$ if and only if

$$r(\mathbf{x}) = \sum_{i=1}^{n} \sum_{j=1}^{m_i} \ell_j^{\Omega_i}(\xi_i^{j_i}) \sum_{\mathbf{k}\in\mathbb{M}_i} \beta_i(j|\mathbf{k}) \prod_{s\in\mathbf{pa}(i)} \ell_{k_s}^{\Omega_s}(\xi_s^{j_s})$$
$$= \sum_{i=1}^{n} \ell_{j_i}^{\Omega_i}(\xi_i^{j_i}) \beta_i(j_i|\{j_s\}_{s\in\mathbf{pa}(i)}) \prod_{s\in\mathbf{pa}(i)} \ell_{j_s}^{\Omega_s}(\xi_s^{j_s}) = \sum_{i=1}^{n} \beta_i(j_i|\{j_s\}_{s\in\mathbf{pa}(i)}) \geq 0.$$

In other words, every variable X_i, together with its parents $\mathbf{pa}(i)$, expresses a degree (positive or negative) $\beta_i(j_i|\{j_s\}_{s\in\mathbf{pa}(i)})$ on $\mathbf{x}$, based only on the values of the i-th variable, $\xi_i^{k_i}$ and its parent values, $\{\xi_s^{k_s}\ \forall s \in \mathbf{pa}(i)\}$. The degrees are summed, and a decision is taken based on the result. The degree expressed by each *coalition* child-parents in the Bayesian network classifier is the logarithm of the ratio between the two probabilities obtained conditioned on the values of the class C,

$$\beta_i(j_i|\{j_s\}_{s\in\mathbf{pa}(i)}) = \ln \frac{P(X_i = \xi_i^{j_i}|X_s(i) = \xi_s^{j_s},\ \forall s \in \mathbf{pa}(i), C = +1)}{P(X_i = \xi_i^{j_i}|X_s(i) = \xi_s^{j_s},\ \forall s \in \mathbf{pa}(i), C = -1)}.$$

3.5 Full Bayesian Network

When the predictor sub-graph is a fully connected Bayesian network (Figure 8), that is, a directed acyclic graph with the maximum number of arcs, we have a fully connected Bayesian network classifier (FBN). A FBN can represent any joint probability distribution over $(C, X_1, \ldots, X_n)$ and so it is a classifier able to induce any decision function over $\Omega = \times_{i=1}^n \Omega_i$ whatsoever. We have that the product of the Lagrange bases, $\prod_{i=1}^n \ell_{k_i}^{\Omega_i}(x_i)$, interpolates the Iverson bracket over all the predictors, that is,

$$\prod_{i=1}^n \ell_{k_i}^{\Omega_i}(x_i) = [x_i = \xi_i^{k_i},\ \forall i = 1, \ldots, n].$$

And so the following lemma holds.

Lemma 11 *If Φ is a classifier for a binary class problem with n categorical predictor variables $X_1, \ldots, X_n$ such that $X_i \in \Omega_i = \{\xi_i^1, \ldots, \xi_i^{m_i}\} \subset \mathbb{R}$, $|\Omega_i| = m_i$, then the associated decision function, f^Φ, is sign-represented by a polynomial of the form*

$$\sum_{\mathbf{k}\in\mathbb{M}} \gamma_{\mathbf{k}} \prod_{i=1}^n \ell_{k_i}^{\Omega_i}(x_i),$$

where $\mathbb{M} = \times_{i=1}^n \{1, \ldots, m_i\}$.

We observe that the coefficients $\gamma_{\mathbf{k}}$ in Lemma 11 are the values of the polynomial at point $(\xi_1^{k_1}, \xi_2^{k_2}, \ldots, \xi_n^{k_n})$, and so $f^\Phi(\xi_1^{k_1}, \xi_2^{k_2}, \ldots, \xi_n^{k_n}) = sgn(\gamma_{\mathbf{k}})$. Roughly speaking, a new instance $(\xi_1^{k_1}, \xi_2^{k_2}, \ldots, \xi_n^{k_n})$ will be classified as $C = +1$ if and only if $\gamma_{\mathbf{k}} > 0$. Moreover the set

$$\mathcal{P}^{FBN} = \left\{ \sum_{\mathbf{k}\in\mathbb{M}} \gamma_{\mathbf{k}} \prod_{i=1}^n \ell_{k_i}^{\Omega_i}(x_i) \text{ s.t. } \gamma_{\mathbf{k}} \in \mathbb{R} \right\}$$

of polynomials, which could sign-represent every classifier, is a space of dimension $M = |\mathbb{M}| = \prod_{i=1}^n m_i$. From now on we will write

$$\delta_{\mathbf{k}}(\mathbf{x}) = \prod_{i=1}^n \ell_{k_i}^{\Omega_i}(x_i), \tag{14}$$

for the $\mathbf{k}$-th element of the canonical basis of $\mathcal{P}^{FBN}$. We call $\{\delta_{\mathbf{k}}\}_{\mathbf{k}\in\Omega}$ the canonical basis because the sign of the coefficients with respect to this basis is the value of the sign-represented decision function. Lemma 11 states that $sgn(\mathcal{P}^{FBN}) = \{-1, 1\}^\Omega$.

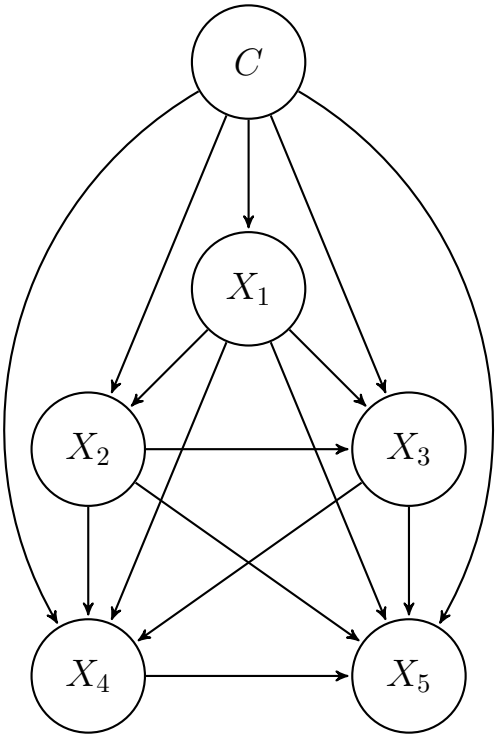


Figure 8: FBN classifier structure with five predictor variables

4. Expressive Power of Bayesian Network Classifiers

So far, we have seen how to build polynomial threshold functions that sign-represent decision functions induced by Bayesian network classifiers. We use now the resulting representation to bound the number of decision functions representable by Bayesian network classifiers. As observed, Lemma 11 says that $sgn(\mathcal{P}^{FBN}) = \{-1,1\}^{\Omega}$. We now study NB, SPODE and BAN through the families of associated polynomial threshold functions. Moreover, we embed those families in $\mathcal{P}^{FBN}$. For predictor variables $X_i \in \Omega_i = \{\xi_i^1, \ldots, \xi_i^{m_i}\}$, $i = 1, \ldots, n$, for every $sp \in \{1, \ldots, n\}$ and a directed acyclic graph $\mathcal{G}$ without V-structures we define

$$\mathcal{P}^{NB} = \left\{ r(\mathbf{x}) = \sum_{i=1}^{n} \left(\sum_{j=1}^{m_i} \alpha_i(j) \ell_j^{\Omega_i}(x_i) \right) \text{ s.t. } \alpha_i(j) \in \mathbb{R} \right\}, \tag{15}$$

$$\mathcal{P}_{sp}^{SPODE} = \left\{ r(\mathbf{x}) = \sum_{i \neq sp} \sum_{j=1}^{m_i} \sum_{k=1}^{m_{sp}} \beta_i(j|k) \ell_k^{\Omega_{sp}}(x_{sp}) \ell_j^{\Omega_i}(x_i) \text{ s.t. } \beta_i(j|k) \in \mathbb{R} \right\}, \tag{16}$$

$$\mathcal{P}_{\mathcal{G}}^{BAN} = \left\{ r(\mathbf{x}) = \sum_{i=1}^{n} \sum_{j=1}^{m_i} \ell_j^{\Omega_i}(x_i) \sum_{\mathbf{k} \in \mathbb{M}_i} \beta_i(j|\mathbf{k}) \prod_{s \in \mathbf{pa}(i)} \ell_{k_s}^{\Omega_s}(x_s) \text{ s.t. } \beta_i(j|\mathbf{k}) \in \mathbb{R} \right\}, \tag{17}$$

where $\mathbf{pa}(i)$ is a function that maps every i into the set of parents of X_i in the directed acyclic graph $\mathcal{G}$, and $\mathbb{M}_i = \times_{s \in \mathbf{pa}(i)} \{1, \ldots, m_s\}$. The families $\mathcal{P}^{NB}$, $\mathcal{P}_{sp}^{SPODE}$ and $\mathcal{P}_{\mathcal{G}}^{BAN}$ are the sets

of polynomials sign-representing the decision functions induced by naive Bayes classifier, SPODE classifier and BAN classifier, respectively. Hence $sgn(\mathcal{P}^{NB})$, $sgn(\mathcal{P}^{SPODE}_{sp})$ and $sgn(\mathcal{P}^{BAN}_{\mathcal{G}})$ are the sets of decision functions induced by naive Bayes, SPODE and BAN classifiers, respectively. Obviously, we have that

$$\mathcal{P}^{NB} \subset \mathcal{P}^{BAN}_{\mathcal{G}} \subset \mathcal{P}^{FBN},$$

and

$$sgn(\mathcal{P}^{NB}) \subset sgn(\mathcal{P}^{BAN}_{\mathcal{G}}) \subset sgn(\mathcal{P}^{FBN}) = \{-1,+1\}^{\Omega}.$$

We can prove that the above sets are indeed subspaces of $\mathcal{P}^{FBN}$ and we can compute their dimensions.

Lemma 12 *$\mathcal{P}^{NB}$ is a subspace of $\mathcal{P}^{FBN}$ of dimension $\sum_{i=1}^n m_i - n + 1$.*

Proof Obviously $\mathcal{P}^{NB} = \left\{ p(\mathbf{x}) = \sum_{i=1}^n \left(\sum_{j=1}^{m_i} \alpha_i(j) \ell_j^{\Omega_i}(x_i) \right), \alpha_i(j) \in \mathbb{R} \right\}$ is a subspace of $\mathcal{P}^{FBN}$. The union of the Lagrange bases over different variables is not a basis, because for each $i = 1, \ldots, n$ we have that

$$1 = \sum_{j=1}^{m_i} \ell_j^{\Omega_i}(x_i) \text{ for every } x_i \in \mathbb{R}.$$

So for every i, we can define

$$\mathcal{B}_i = \left\{ \bigcup_{j=2}^{m_i} \{ l_j^{\Omega_i}(x_i) \} \right\} \cup \{e_0\},$$

where e_0 is the polynomial constant 1, and we find that $\mathcal{B}_i$ is a basis of polynomials in x_i of degree $|\Omega_i| - 1 = m_i - 1$, equivalent to the Lagrange basis over Ω_i. Then, we have that

$$\mathcal{B} = \bigcup_{i=1}^n \mathcal{B}_i = \bigcup_{i=1}^n \bigcup_{j=2}^{m_i} \left\{ l_j^{\Omega_i}(x_i) \right\} \cup \{e_0\}$$

generates the subspace $\mathcal{P}^{NB}$. We prove that $\mathcal{B}$ is in fact a basis of $\mathcal{P}^{NB}$. We have to prove that the elements of $\mathcal{B}$ are linearly independent. We consider

$$p(x_1, x_2, \ldots, x_n) = \sum_{i=1}^n \sum_{j=2}^{m_i} \alpha_i(j) \ell_j^{\Omega_i}(x_i) + \alpha_0 e_0 = 0, \, \forall (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n.$$

If, as usual, $\Omega_i = \{\xi_i^1, \ldots, \xi_i^{m_i}\}$, let us consider $p(x_1, \ldots, x_n)$ evaluated in $(\xi_1^1, \xi_2^1, \ldots, \xi_n^1)$,

$$0 = p(\xi_1^1, \xi_2^1, \ldots, \xi_n^1) = \sum_{i=1}^n \sum_{j=2}^{m_i} \alpha_i(j) \ell_j^{\Omega_i}(\xi_i^1) + \alpha_0 e_0 = \alpha_0,$$

since $\ell_j^{\Omega_i}(\xi_i^1) = 0$ for every $j \neq 1$. And so $\alpha_0 = 0$. We now evaluate $p(\cdot)$ over $(\xi_1^j, \xi_2^1, \ldots, \xi_n^1)$ and we have that, for every $j = 2, \ldots, m_i$,

$$0 = p(\xi_1^j, \xi_2^1, \ldots, \xi_n^1) = \alpha_1(j),$$

since $\ell_j^{\Omega_1}(\xi_1^j) = 1$ for every $j = 2, \ldots, m_1$. We repeat the above argument for every variable x_i, $i = 1, \ldots, n$ and we obtain $\alpha_i(j) = 0$ for every $i = 1, \ldots, n$ and every $j = 2, \ldots, m_i$. We have proved that the elements of $\mathcal{B}$ generate $\mathcal{P}^{NB}$ and are linearly independent, so they form a basis of $\mathcal{P}^{NB}$. Consequently we obtain

$$\dim(\mathcal{P}^{NB}) = |\mathcal{B}| = \sum_{i=1}^{n} m_i - n + 1.$$

■

Analogously we can prove, in the general case,

Lemma 13 *For every Bayesian network classifier without V-structures in the predictor sub-graph $\mathcal{G}$, the set $\mathcal{P}_{\mathcal{G}}^{BAN}$ is a subspace of $\mathcal{P}^{FBN}$ of dimension*

$$\sum_{i=1}^{n} \left((m_i - 1) \prod_{s \in \mathbf{pa}(i)} m_s \right) + 1.$$

And, in the particular case of $SPODE$, we have,

Lemma 14 *For every $sp = 1, \ldots, n$, the set $\mathcal{P}_{sp}^{SPODE}$ is a subspace of $\mathcal{P}^{FBN}$ of dimension $m_{sp}\left(1 - n + \sum_{i \neq sp} m_i\right)$.*

We now consider the space $\mathcal{P}^{FBN}$ with respect to the canonical basis given by Equation (14). With respect to this coordinate system we have that each orthant represents a decision function. We know that the number of orthants of an M-dimensional space is 2^M, the number of decision functions over a set of cardinality M. Since we now have a bijection between orthants in $\mathcal{P}^{FBN}$ and decision functions over Ω, in order to compute how many decision functions are representable by a class of Bayesian network classifier (NB, SPODE or BAN) we merely have to count the number of orthants in $\mathcal{P}^{FBN}$ intersected by the corresponding subspaces ($\mathcal{P}^{NB}$, $\mathcal{P}_{sp}^{SPODE}$, $\mathcal{P}_{\mathcal{G}}^{BAN}$).

Theorem 15 (Flatto, 1970) *A d-dimensional subspace in an M-dimensional space intersects at most $C(M, d) = 2\sum_{k=0}^{d-1} \binom{M-1}{k}$ orthants with equality if and only if it is in general position.*

Definition 16 *A d-dimensional subspace V of $\mathbb{R}^M$ is in general position if the M subspaces $V \cap H_i$, where $H_i = \{\mathbf{x} \in \mathbb{R}^n \text{ s.t. } x_i = 0\}$ are hyperplanes of V in general position, that is, all the intersections of d of such hyperplanes are the zero vector. Precisely, for all $J \subset \{1, \ldots, M\}$ such that $|J| = d$ we have that $\bigcap_{j \in J} (V \cap H_j) = \mathbf{0}$.*

Applying Theorem 15 to our case, we find that the space $\mathcal{P}^{FBN}$ is minimal in the following sense.

Corollary 17 *If V is a d-dimensional subspace of $\mathcal{P}^{FBN}$, then $|sgn(V)| \leq C(M, d)$, where $M = dim(\mathcal{P}^{FBN})$ and equality holds if and only if V is in general position with respect to the canonical basis of $\mathcal{P}^{FBN}$.*

As a first result of Corollary 17 we have that the space $\mathcal{P}^{FBN}$ is the *smallest* vector space of polynomials in $x_1, \ldots, x_n$ that sign-represents every decision function over Ω, that is, there is not a space V of polynomials in $x_1, \ldots, x_n$ with degrees in each variable x_i that are less or equal than $m_i - 1$ such that $sgn(V) = \{-1, +1\}^{\Omega}$ and $dim(V) < dim(\mathcal{P}^{FBN})$. This justifies the choice of $\mathcal{P}^{FBN}$ as the space to study the polynomial families defined in Equations (15), (16) and (17). Next, we can use Corollary 17 combined with Lemma 13 to upper bound the number of decision functions that are sign-representable by BAN classifiers with a fixed predictor sub-graph $\mathcal{G}$ not containing V-structures.

Corollary 18 *Consider a BAN classifier over predictor variables $X_i \in \Omega_i$, $|\Omega_i| = m_i$ for every $i = 1, \ldots, n$. Moreover suppose that the predictor sub-graph $\mathcal{G}$ does not contain V-structures. Then we have*

$$2^d \leq |sgn(\mathcal{P}_{\mathcal{G}}^{BAN})| \leq C(M, d) = 2 \sum_{k=0}^{d-1} \binom{M-1}{k},$$

where $d = \sum_{i=1}^{n} \left((m_i - 1) \prod_{s \in \mathbf{pa}(i)} m_s \right) + 1$ and $M = \prod_{i=1}^{n} m_i$.

Peot (1996) observed that naive Bayes could only represent a fraction of dichotomies (binary decision) on binary predictors, and that this fraction goes to zero as the number of predictors increase, we extend this observation to BAN classifier without V-structures as follows.

Corollary 19 *We consider, for every $n \in \mathbb{N}$, classification problems with predictors $X_i \in \Omega_i \subset \mathbb{R}$, $|\Omega_i| = m_i$ for $i = 1, \ldots, n$. For every n, let $\mathcal{G}_n$ be a directed acyclic graph over the predictor variables, not containing V-structures. Suppose moreover that if $\mathbf{pa}_n(i)$ are the functions that map every X_i into the set of parents in the graph $\mathcal{G}_n$,*

$$|\mathbf{pa}_n(i)| \leq K \quad \forall n \in \mathbb{N} \text{ and } i \in \{1, \ldots, n\},$$

then we have that

$$\lim_{n \to \infty} \frac{\left| sgn\left(P_{\mathcal{G}_n}^{BAN}\right) \right|}{\left| \{-1, +1\}^{\Omega(n)} \right|} = \lim_{n \to \infty} \frac{\left| sgn\left(P_{\mathcal{G}_n}^{BAN}\right) \right|}{2^{|\Omega(n)|}} = 0,$$

where $\Omega(n) = \times_{i=1}^{n} \Omega_i$. In other words, the fraction of decision functions representable by BAN classifiers, with a fixed maximum number of parents for each variable, becomes vanishingly small by increasing the number of predictors.

Proof For every $n \in \mathbb{N}$, we apply Corollary 18 and we obtain

$$\left| sgn\left(\mathcal{P}_{\mathcal{G}_n}^{BAN}\right) \right| \leq C\left(M(n), d(n)\right) = 2 \sum_{k=0}^{d(n)-1} \binom{M(n)-1}{k},$$

where $d(n) = \sum_{i=1}^{n} \left((m_i - 1) \prod_{s \in \mathbf{pa}(i)} m_s \right) + 1$ and $M(n) = |\Omega(n)| = \prod_{i=1}^{n} m_i$. We observe now that, as $n \to \infty$,

$$\frac{d(n)}{M(n)} \to 0$$

and thus,

$$\frac{C(M(n), d(n))}{2^{M(n)}} \to 0,$$

which proves the statement. ■

5. Conclusions

In this paper we have shown how to build polynomial threshold functions related to Bayesian network classifiers. Our results reveal connections between the algebraic structure of the decision functions induced by BN classifiers and the topology of the structure of the predictor sub-graph. In absence of V-structures in the predictor sub-graph we have also proved that the specific polynomial representation fully characterized the type of Bayesian network classifier. By representing classifiers by polynomial threshold functions, we can obtain bounds on the number of decision functions which can be induced by Bayesian network classifiers with a given structure. The bounding does not hold in presence of V-structures in the predictor sub-graph. Strong characterizations of induced decision functions cannot be proven due to the conditional independence of V-structure. Moreover we observe that the obtained polynomial representation permits to easily prove the results of Ling and Zhang (2002) for BAN classifiers without V-structures.

The bounds points to the fact, already conjectured by Peot (1996) for naive Bayes, that if we fix the maximum number of parents in a Bayesian network classifier, the type of classifier considered is not *scalable*, in other words, more complex classifiers are expected to perform better when dealing with a large number of predictor variables.

Moreover, the resulting bounds for the number of decision functions representable are strictly upper bounds since the subspaces generated by the different Bayesian networks considered are not in general position. What happens in the case of subspaces not in general position? Clearly we have to define some other property to characterize the *position* of a subspace with respect to orthants in some given basis and try to count the number of such intersected orthants. With similar geometric results we will be able to precisely count the number of decision functions representable by a given Bayesian network classifier, and we will be able to compute the gain in expressivity from simple to more complicated Bayesian network classifiers.

Acknowledgments

The authors thank the anonymous reviewers for their constructive comments and corrections. This research has been partially supported by the Spanish Ministry of Economy and Competitiveness through Cajal Blue Brain (C080020-09) and TIN2013-41592-P projects and by the Madrid Regional government through S2013/ICE-2845-CASI-CAM-CM project.

References

Milton Abramowitz and Irene A. Stegun. *Handbook of Mathematical Functions: With Formulas, Graphs, and Mathematical Tables*. Applied Mathematics Series. Dover Publications, 1964.

Concha Bielza and Pedro Larrañaga. Discrete Bayesian network classifiers: A survey. *ACM Comput. Surv.*, 47(1):5:1–5:43, 2014.

Pedro Domingos and Michael Pazzani. On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning*, 29(2-3):103–130, 1997.

Leopold Flatto. A new proof of the transposition theorem. *Proceedings of the American Mathematical Society*, 24(1):29–31, 1970.

Nir Friedman, Dan Geiger, and Moises Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29(2-3):131–163, 1997.

Mark Hall. A decision tree-based attribute weighting filter for naive Bayes. In Max Bramer, Frans Coenen, and Andrew Tuson, editors, *Research and Development in Intelligent Systems XXIII*, pages 59–70. Springer London, 2007.

Kenneth E. Iverson. *A Programming Language*. John Wiley & Sons, Inc., New York, 1962.

Manfred Jaeger. Probabilistic classifiers and the concepts they recognize. In Tom Fawcett and Nina Mishra, editors, *Proceedings of the Twentieth International Conference on Machine Learning (ICML-03)*, pages 266–273. AAAI Press, 2003.

Harold Jeffreys and Bertha Jeffreys. *Methods of Mathematical Physics*. Cambridge Mathematical Library. Cambridge University Press, 1999.

Eamonn J. Keogh and Michael J. Pazzani. Learning the structure of augmented Bayesian classifiers. *International Journal on Artificial Intelligence Tools*, 11(04):587–601, 2002.

Charles X. Ling and Huajie Zhang. The representational power of discrete Bayesian networks. *Journal of Machine Learning Research*, 3:709–721, 2002.

Marvin Minsky. Steps toward artificial intelligence. In *Computers and Thought*, pages 406–450. McGraw-Hill, 1961.

Atsuyoshi Nakamura, Michael Schmitt, Niels Schmitt, and Hans Ulrich Simon. Inner product spaces for Bayesian networks. *Journal of Machine Learning Research*, 6:1383–1403, 2005.

Ryan O'Donnell and Rocco A. Servedio. New degree bounds for polynomial threshold functions. *Combinatorica*, 30(3):327–358, 2010.

Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., 1988.

Mark A. Peot. Geometric implications of the naive Bayes assumption. In *Proceedings of the Twelfth International Conference on Uncertainty in Artificial Intelligence*, UAI'96, pages 414–419, San Francisco, 1996. Morgan Kaufmann Publishers Inc.

Giovanni Pistone, Eva Riccomagno, and Henry P. Wynn. Gröbner bases and factorisation in discrete probability and Bayes. *Statistics and Computing*, 11(1):37–46, 2001.

Vladimir N. Vapnik and Alexy Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and Its Applications*, 16 (2):264–280, 1971.

Chi Wang and A.C. Williams. The threshold order of a Boolean function. *Discrete Applied Mathematics*, 31(1):51–69, 1991.

Geoffrey I. Webb and Michael J. Pazzani. Adjusted probability naive Bayesian induction. In *Proceedings of the Eleventh Australian Joint Conference on Artificial Intelligence*, pages 285–295. Springer-Verlag, 1998.

Youlong Yang and Yan Wu. On the properties of concept classes induced by multivalued Bayesian networks. *Information Sciences*, 184(1):155–165, 2012.

Nayyar A. Zaidi, Jesus Cerquides, Mark J. Carman, and Geoffrey I. Webb. Alleviating naive Bayes attribute independence assumption by attribute weighting. *Journal of Machine Learning Research*, 14:1947–1988, 2013.

Journal of Machine Learning Research 16 (2015) 2751-2795 Submitted 7/12; 12/14; Published 12/15

A View of Margin Losses as Regularizers of Probability Estimates

Hamed Masnadi-Shirazi HMASNADI@SHIRAZU.AC.IR
School of Electrical and Computer Engineering,
Shiraz University,
Shiraz, Iran

Nuno Vasconcelos NUNO@UCSD.EDU
Statistical Visual Computing Laboratory,
University of California, San Diego
La Jolla, CA 92039, USA

Editor: Saharon Rosset

Abstract

Regularization is commonly used in classifier design, to assure good generalization. Classical regularization enforces a cost on classifier complexity, by constraining parameters. This is usually combined with a margin loss, which favors large-margin decision rules. A novel and unified view of this architecture is proposed, by showing that margin losses act as regularizers of posterior class probabilities, in a way that amplifies classical parameter regularization. The problem of controlling the regularization strength of a margin loss is considered, using a decomposition of the loss in terms of a link and a binding function. The link function is shown to be responsible for the regularization strength of the loss, while the binding function determines its outlier robustness. A large class of losses is then categorized into equivalence classes of identical regularization strength or outlier robustness. It is shown that losses in the same regularization class can be parameterized so as to have tunable regularization strength. This parameterization is finally used to derive boosting algorithms with loss regularization (BoostLR). Three classes of tunable regularization losses are considered in detail. Canonical losses can implement all regularization behaviors but have no flexibility in terms of outlier modeling. Shrinkage losses support equally parameterized link and binding functions, leading to boosting algorithms that implement the popular shrinkage procedure. This offers a new explanation for shrinkage as a special case of loss-based regularization. Finally, α-tunable losses enable the independent parameterization of link and binding functions, leading to boosting algorithms of great flexibility. This is illustrated by the derivation of an algorithm that generalizes both AdaBoost and LogitBoost, behaving as either one when that best suits the data to classify. Various experiments provide evidence of the benefits of probability regularization for both classification and estimation of posterior class probabilities.

Keywords: classification, margin losses, regularization, boosting, probability elicitation, generalization, loss functions, link functions, binding functions, shrinkage

1. Introduction

The ability to generalize beyond the training set is a central challenge for classifier design. A binary classifier is usually implemented by thresholding a continuous function, the classifier predictor, of a high-dimensional feature vector. Predictors are frequently affine functions, whose level sets (decision boundaries) are hyperplanes in feature space. Optimal predictors minimize the empirical expectation of a loss function, or risk, on a training set. Modern risks guarantee good generalization by enforcing large margins and parameter regularization. Large margins follow from the use of margin losses, such as the hinge loss of the support vector machine (SVM), the exponential loss of AdaBoost, or the logistic loss of logistic regression and LogitBoost. These are all upper-bounds on the zero-one classification loss of classical Bayes decision theory. Unlike the latter, margin losses assign a penalty to examples correctly classified but close to the boundary. This guarantees a classification margin and improved generalization (Vapnik, 1998). Regularization is implemented by penalizing predictors with many degrees of freedom. This is usually done by augmenting the risk with a penalty on the norm of the parameter vector. Under a Bayesian interpretation of risk minimization, different norms correspond to different priors on predictor parameters, which enforce different requirements on the sparseness of the optimal solution.

While for some popular classifiers, e.g. the SVM, regularization is a natural side-product of risk minimization under a margin loss (Moguerza and Munoz, 2006; Chapelle, 2007; Huang et al., 2014), the relation between the two is not always as clear for other learning methods, e.g. boosting. Regularization can be added to boosting (Buhlmann and Hothorn, 2007; Lugosi and Vayatis, 2004; Blanchard et al., 2003) in a number of ways, including restricting the number of boosting iterations (Raskutti et al., 2014; Natekin and Knoll, 2013; Zhang and Yu, 2005; Rosset et al., 2004; Jiang, 2004; Buhlmann and Yu, 2003), adding a regularization term (Saha et al., 2013; Culp et al., 2011; Xiang et al., 2009; Bickel et al., 2006; Xi et al., 2009), restricting the weight update rule (Lozano et al., 2014, 2006; Lugosi and Vayatis, 2004; Jin et al., 2003) or using divergence measures (Liu and Vemuri, 2011) and has been implemented for both the supervised and semi-supervised settings (Chen and Wang, 2008, 2011). However, many boosting algorithms lack explicit parameter regularization. Although boosting could eventually overfit (Friedman et al., 2000; Rosset et al., 2004), and there is an implicit regularization when the number of boosting iterations is limited (Raskutti et al., 2014; Natekin and Knoll, 2013; Zhang and Yu, 2005; Rosset et al., 2004; Jiang, 2004; Buhlmann and Yu, 2003), there are several examples of successful boosting on very high dimensional spaces, using complicated ensembles of thousands of weak learners, and no explicit regularization (Viola and Jones, 2004; Schapire and Singer, 2000; Viola et al., 2003; Wu and Nevatia, 2007; Avidan, 2007). This suggests that regularization is somehow implicit in large margins, and additional parameter regularization may not always be critical, or even necessary. In fact, in domains like computer vision, large margin classifiers are more popular than classifiers that enforce regularization but not large margins, e.g. generative models with regularizing priors. This suggests that the regularization implicit in large margins is complementary to parameter regularization. However, this connection has not been thoroughly studied in the literature.

In this work, we approach the problem by studying the properties of margin losses. This builds on prior work highlighting the importance of three components of risk mini-

mization: the loss ϕ, the minimum risk C^*_ϕ, and a link function f^*_ϕ that maps posterior class probabilities to classifier predictions (Friedman et al., 2000; Zhang, 2004; Buja et al., 2006; Masnadi-Shirazi and Vasconcelos, 2008; Reid and Williamson, 2010). We consider the subset of losses of invertible link, since this enables the recovery of class posteriors from predictor outputs. Losses with this property are known as proper losses and important for applications that require estimates of classification confidence, e.g. multiclass decision rules based on binary classifiers (Zadrozny, 2001; Rifkin and Klautau, 2004; Gonen et al., 2008; Shiraishi and Fukumizu, 2011). We provide a new interpretation of these losses as regularizers of finite sample probability estimates and show that this regularization has at least two important properties for classifier design. First, it combines multiplicatively with classical parameter regularization, amplifying it in a way that tightens classification error bounds. Second, probability regularization strength is proportional to loss margin for a large class of link functions, denoted generalized logit links. This enables the introduction of tunable regularization losses ϕ_σ, parameterized by a probability regularization gain σ. A procedure to derive boosting algorithms of tunable loss regularization (BoostLR) from these losses is also provided. BoostLR algorithms generalize the GradientBoost procedure (Friedman, 2001), differing only in the example weighting mechanism, which is determined by the loss ϕ_σ.

To characterize the behavior of these algorithms, we study the space $\mathcal{R}$ of proper losses ϕ of generalized logit link. It is shown that any such ϕ is uniquely defined by two components: the link f^*_ϕ and a binding function β_ϕ that maps f^*_ϕ into the minimum risk C^*_ϕ. This decomposition has at least two interesting properties. First, the two components have a functional interpretation: while f^*_ϕ determines the probability regularization strength of ϕ, β_ϕ determines its robustness to outliers. Second, both β_ϕ and f^*_ϕ define equivalence classes in $\mathcal{R}$. It follows that $\mathcal{R}$ can be partitioned into subsets of losses that have either the same outlier robustness or probability regularization properties. It is shown that the former are isomorphic to a set of symmetric scale probability density functions and the latter to the set of monotonically decreasing odd functions. Three loss classes, with three different binding functions, are then studied in greater detail. The first, the class of canonical losses, consists of losses of linear binding function. This includes some of the most popular losses in the literature, e.g. the logistic. While they can implement all possible regularization behaviors, these losses have no additional degrees of freedom. In this sense, they are the simplest tunable regularization losses. This simplicity enables a detailed analytical characterization of their shape and how this shape is affected by the regularization gain. The second, the class of shrinkage losses, is a superset of the class of canonical losses. Unlike their canonical counterparts, shrinkage losses support nonlinear binding functions, and thus more sophisticated handling of outliers. However, they require an identical parameterization of the link and binding function. It is shown that, under this constraint, BoostLR implements the popular shrinkage regularization procedure (Hastie et al., 2001). Finally, the class of α-tunable losses enables independent parameterization of the link and binding functions. This endows the losses in this class, and the associated BoostLR algorithms, with a great deal of flexibility. We illustrate this by introducing an α-tunable loss that generalizes both the exponential loss of AdaBoost and the logistic loss of LogitBoost, allowing BoostLR to behave as either of the two algorithms, so as to best suit the data to classify.

The paper is organized as follows. Section 2 briefly reviews classifier design by risk minimization. The view of margin losses as regularizers of probability estimates is introduced in Section 3. Section 4 characterizes the regularization strength of proper losses of generalized logit link. Tunable regularization losses and binding functions are introduced in Section 5, which also introduces the BoostLR algorithm. The structure of $\mathcal{R}$ is then characterized in Section 6, which introduces canonical, shrinkage, and α-tunable losses. An extensive set of experiments on various aspects of probability regularization is reported in Section 7. Finally, some conclusions are drawn in Section 8.

2. Loss Functions and Risk Minimization

We start by reviewing the principles of classifier design by risk minimization (Friedman et al., 2000; Zhang, 2004; Buja et al., 2006; Masnadi-Shirazi and Vasconcelos, 2008) .

2.1 The Classification Problem

A classifier h maps a feature vector $\mathbf{x} \in \mathcal{X}$ to a class label $y \in \{-1, 1\}$, according to

$$h(\mathbf{x}) = sign[p(\mathbf{x})], \tag{1}$$

where $p : \mathcal{X} \to \mathbb{R}$ is the classifier predictor. Feature vectors and class labels are drawn from probability distributions $P_{\mathbf{X}}(\mathbf{x})$ and $P_{Y|X}(y|\mathbf{x})$ respectively. Given a non-negative loss function $L(\mathbf{x}, y)$, the optimal predictor $p^*(\mathbf{x})$ minimizes the risk

$$R(p) = E_{\mathbf{X},Y}[L(p(\mathbf{x}), y)]. \tag{2}$$

This is equivalent to minimizing the conditional risk

$$E_{Y|\mathbf{X}}[L(p(\mathbf{x}), y)|\mathbf{X} = \mathbf{x}]$$

for all $\mathbf{x} \in \mathcal{X}$. It is frequently useful to express $p(\mathbf{x})$ as a composition of two functions,

$$p(\mathbf{x}) = f(\eta(\mathbf{x})),$$

where $\eta(\mathbf{x}) = P_{Y|\mathbf{X}}(1|\mathbf{x})$ is the posterior probability function, and $f : [0, 1] \to \mathbb{R}$ a link function. The problem of learning the optimal predictor can thus be decomposed into the problems of learning the optimal link $f^*(\eta)$ and estimating the posterior function $\eta(\mathbf{x})$. Since $f^*(\eta)$ can usually be determined analytically, this reduces to estimating $\eta(\mathbf{x})$, whenever $f^*(\eta)$ is a one-to-one mapping.

In classical statistics, learning is usually based on the zero-one loss

$$L_{0/1}(y, p) = \frac{1 - sign(yp)}{2} = \begin{cases} 0, & \text{if } y = sign(p); \\ 1, & \text{if } y \neq sign(p), \end{cases}$$

where we omit the dependence on $\mathbf{x}$ for notational simplicity. The associated conditional risk

$$C_{0/1}(\eta, p) = \eta\frac{1 - sign(p)}{2} + (1 - \eta)\frac{1 + sign(p)}{2} = \begin{cases} 1 - \eta, & \text{if } p = f(\eta) \geq 0; \\ \eta, & \text{if } p = f(\eta) < 0, \end{cases}$$

is the probability of error of the classifier of (1), and is minimized by any f^* such that

$$\begin{cases} f^*(\eta) > 0 & \text{if } \eta > \frac{1}{2} \\ f^*(\eta) = 0 & \text{if } \eta = \frac{1}{2} \\ f^*(\eta) < 0 & \text{if } \eta < \frac{1}{2}. \end{cases} \tag{3}$$

The optimal classifier $h^*(\mathbf{x}) = sign[p^*(\mathbf{x})]$, where $p^* = f^*(\eta)$, is the well known Bayes decision rule, and has minimum conditional (zero-one) risk

$$\begin{aligned} C^*_{0/1}(\eta) &= \eta \left(\frac{1}{2} - \frac{1}{2} sign(2\eta - 1) \right) + (1 - \eta) \left(\frac{1}{2} + \frac{1}{2} sign(2\eta - 1) \right) \\ &= \min\{\eta, 1 - \eta\}. \end{aligned}$$

2.2 Learning from Finite Samples

Practical learning algorithms produce an estimate $\hat{p}^*(\mathbf{x})$ of the optimal predictor by minimizing an empirical estimate of (2), the empirical risk, from a training sample $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$

$$R_{emp}(p) = \frac{1}{n} \sum_i L(p(\mathbf{x}_i), y_i). \tag{4}$$

This can be formulated as fitting a model $\hat{\eta}(\mathbf{x}) = [f^*]^{-1}(p(\mathbf{x}; \mathbf{w}))$ to the sample $\mathcal{D}$, where f^* is an invertible link that satisfies (3) and $p(\mathbf{x}; \mathbf{w})$ a parametric predictor. Two commonly used links are

$$f^* = 2\eta - 1 \quad \text{and} \quad f^* = \log \frac{\eta}{1 - \eta}.$$

In this way, the learning problem is reduced to the estimation of the model parameters $\mathbf{w}$ of minimum empirical risk. Most modern learning techniques rely on a linear predictor, implemented on either $\mathcal{X}$ - $p(\mathbf{x}, \mathbf{w}) = \mathbf{w}^T\mathbf{x}$ - or some transformed space - $p(\mathbf{x}, \mathbf{w}) = \mathbf{w}^T\Phi(\mathbf{x})$. For example, logistic regression (Hosmer and Lemeshow, 2000) uses the logit link $f^* = \log \frac{\eta}{1-\eta}$, or equivalently the logistic inverse link $[f^*]^{-1}(v) = \frac{e^v}{1+e^v}$, and learns a linear predictor $p(\mathbf{x}, \mathbf{w}) = \mathbf{w}^T\mathbf{x}$. When a transformation $\Phi(\mathbf{x})$ is used, it is either implemented indirectly with recourse to a kernel function, e.g. kernelized logistic regression (Zhu and Hastie, 2001), or learned. For example, boosting algorithms rely on a transformation $\Phi(\mathbf{x}) = (h_1(\mathbf{x}), \ldots, h_m(\mathbf{x}))$ where $h_i(\mathbf{x})$ is a weak or base classifier selected during training. In this case, the predictor has the form

$$p(\mathbf{x}; \mathbf{w}) = \sum_i w_i h_i(\mathbf{x}). \tag{5}$$

In all cases, given the optimal predictor estimate $\hat{p}^*(\mathbf{x}) = p(\mathbf{x}, \mathbf{w}^*)$, estimates of the posterior probability $\eta(\mathbf{x})$ can be obtained with $\hat{\eta}(\mathbf{x}) = [f^*]^{-1}(\hat{p}^*(\mathbf{x}))$. However, when learning is based on the empirical risk of (4), convergence to the true probabilities is only guaranteed asymptotically and for certain loss functions $L(.,.)$. Even when this is the case, learning algorithms can easily overfit to the training set, for finite samples. The minimum of (4) is achieved for some empirical predictor

$$\hat{p}^*(\mathbf{x}) = p^*(\mathbf{x}) + \epsilon_p(\mathbf{x}), \tag{6}$$

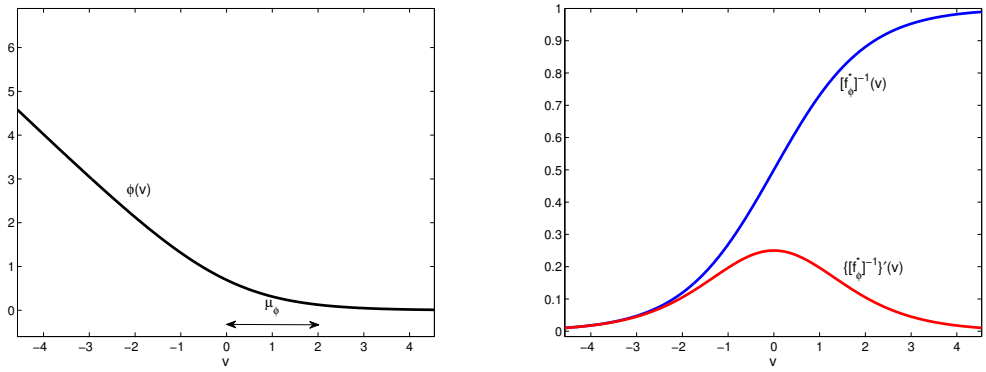


Figure 1: Left: A margin loss function (the logistic loss) of margin parameter μ_ϕ, defined in (25). Right: corresponding inverse link (in blue) and its growth rate (in red).

where $p^*(\mathbf{x})$ is the optimal predictor and $\epsilon_p(\mathbf{x})$ a prediction error, sampled from a zero mean distribution of decreasing variance with sample size. For a given sample size, a predictor with error of smaller variance is said to generalize better. One popular mechanism to prevent overfitting is to regularize the parameter vector $\mathbf{w}$, by imposing a penalty on its norm, i.e. minimizing

$$R_{emp}(p) = \frac{1}{n}\sum_i L(p(\mathbf{x}_i), y_i) + \lambda||\mathbf{w}||_l$$

instead of (4). We refer to this as parameter regularization.

2.3 Margin Losses

Another possibility is to change the loss function, e.g. by replacing the 0-1 loss with a margin loss $L_\phi(y, p(\mathbf{x})) = \phi(yp(\mathbf{x}))$. As illustrated in Figure 1 (left), these losses assign a non-zero penalty to small positive values of the margin yp, i.e. in the range $0 < yp < \mu_\phi$, where μ_ϕ is a parameter, denoted the loss margin. Commonly used margin losses include the exponential loss of AdaBoost, the logistic loss (shown in the figure) of logistic regression, and the hinge loss of SVMs. The resulting large-margin classifiers have better finite sample performance (generalization) than those produced by the 0-1 loss (Vapnik, 1998). The associated conditional risk

$$C_\phi(\eta, p) = C_\phi(\eta, f(\eta)) = \eta\phi(f(\eta)) + (1-\eta)\phi(-f(\eta)) \tag{7}$$

is minimized by the link

$$f^*_\phi(\eta) = \arg\min_f C_\phi(\eta, f) \tag{8}$$

leading to the minimum conditional risk function

$$C^*_\phi(\eta) = C_\phi(\eta, f^*_\phi). \tag{9}$$

Algorithm	$\phi(v)$	$f^*_\phi(\eta)$	$[f^*_\phi]^{-1}(v)$	$C^*_\phi(\eta)$
SVM	$\max(1-v,0)$	$sign(2\eta-1)$	NA	$1-\lvert 2\eta-1\rvert$
Boosting	$\exp(-v)$	$\frac{1}{2}\log\frac{\eta}{1-\eta}$	$\frac{e^{2v}}{1+e^{2v}}$	$2\sqrt{\eta(1-\eta)}$
Logistic Regression	$\log(1+e^{-v})$	$\log\frac{\eta}{1-\eta}$	$\frac{e^{v}}{1+e^{v}}$	$-\eta\log\eta-(1-\eta)\log(1-\eta)$

Table 1: Loss ϕ, optimal link $f^*_\phi(\eta)$, optimal inverse link $[f^*_\phi]^{-1}(v)$, and minimum conditional risk $C^*_\phi(\eta)$ of popular learning algorithms.

Unlike the 0-1 loss, the optimal link is usually unique for margin losses and computable in closed-form, by solving $\eta\phi'(f^*_\phi(\eta)) = (1-\eta)\phi'(-f^*_\phi(\eta))$ for f^*_ϕ. Table 1 lists the loss, optimal link, and minimum risk of popular margin losses.

The adoption of a margin loss can be equivalent to the addition of parameter regularization. For example, a critical step of the SVM derivation is a normalization that makes the margin identical to $1/||\mathbf{w}||$, where $\mathbf{w}$ is the normal of the SVM hyperplane $p(\mathbf{x};\mathbf{w}) = \mathbf{w}^T\mathbf{x}$ (Moguerza and Munoz, 2006; Chapelle, 2007). This renders margin maximization identical to the minimization of hyperplane norm, leading to the interpretation of the SVM as minimizing the hinge loss under a regularization constraint on $\mathbf{w}$ (Moguerza and Munoz, 2006; Chapelle, 2007), i.e.

$$R_{SVM}(\mathbf{w}) = \frac{1}{n}\sum_i \max[0, 1 - yp(\mathbf{x}_i;\mathbf{w})] + \lambda||\mathbf{w}||^2. \tag{10}$$

In this case, larger margins translate directly into the regularization of classifier parameters. This does not, however, hold for all large margin learning algorithms. For example, boosting does not use explicit parameter regularization, although regularization is implicit in early stopping (Raskutti et al., 2014; Natekin and Knoll, 2013; Zhang and Yu, 2005; Rosset et al., 2004; Jiang, 2004; Buhlmann and Yu, 2003). This consists of terminating the algorithm after a small number of iterations. While many bounds have been derived to characterize the generalization performance of large margin classifiers, it is not always clear how much of the generalization ability is due to the loss vs. parameter regularization. In what follows, we show that margin losses can themselves be interpreted as regularizers. However, instead of regularizing predictor parameters, they directly regularize posterior probability estimates, by acting on the predictor output. This suggests a complementary role for loss-based and parameter regularization. We will see that the two types of regularization in fact have a multiplicative effect.

3. Proper Losses and Probability Regularization

We start by discussing the role of margin losses as probability regularizers.

3.1 Regularization Losses

For any margin loss whose link of (8) is invertible, posterior probabilities can be recovered from

$$\eta(\mathbf{x}) = [f_\phi^*]^{-1}(p^*(\mathbf{x})). \tag{11}$$

Whenever this is the case, the loss is said to be proper[1] and the predictor calibrated (DeGroot and Fienberg, 1983; Platt, 2000; Niculescu-Mizil and Caruana, 2005; Gneiting and Raftery, 2007). For finite samples, estimates of the probabilities $\eta(\mathbf{x})$ are obtained from the empirical predictor $\hat{p}^*$ with

$$\hat{\eta}(\mathbf{x}) = [f_\phi^*]^{-1}(\hat{p}^*(\mathbf{x})). \tag{12}$$

Parameter regularization improves estimates $\hat{p}^*(\mathbf{x})$ by constraining predictor parameters. For example, a linear predictor estimate $\hat{p}^*(\mathbf{x};\hat{\mathbf{w}}) = \hat{\mathbf{w}}^T\mathbf{x}$ can be written in the form of (6), with $p^*(\mathbf{x}) = \mathbf{w}^{*T}\mathbf{x}$ and $\epsilon_p(\mathbf{x}) = \mathbf{w}_\epsilon{}^T\mathbf{x}$, where $\mathbf{w}_\epsilon$ is a parameter estimation error. The regularization of (10) reduces $\mathbf{w}_\epsilon$ and the prediction error $\epsilon_p(\mathbf{x})$, improving probability estimates in (12).

Loss-based regularization complements parameter regularization, by regularizing the probability estimates directly. To see this note that, whenever the loss is proper and the noise component ϵ_p of (6) has small amplitude, (12) can be approximated by its Taylor series expansion around p^*

$$\begin{aligned}\hat{\eta}(\mathbf{x}) &\approx [f_\phi^*]^{-1}(p^*(\mathbf{x})) + \{[f_\phi^*]^{-1}\}'(p^*(\mathbf{x}))\epsilon_p(\mathbf{x}) \\ &= \eta(\mathbf{x}) + \epsilon_\eta(\mathbf{x})\end{aligned}$$

with

$$\epsilon_\eta(\mathbf{x}) = \{[f_\phi^*]^{-1}\}'(p^*(\mathbf{x}))\epsilon_p(\mathbf{x}). \tag{13}$$

If $|\{[f_\phi^*]^{-1}\}'(p^*(\mathbf{x}))| < 1$ the probability estimation noise ϵ_η has smaller magnitude than the prediction noise ϵ_p. Hence, for equivalent prediction error ϵ_p, a loss ϕ with inverse link $[f_\phi^*]^{-1}$ of smaller growth rate $|\{[f_\phi^*]^{-1}\}'(v)|$ produces more accurate probability estimates. Figure 1 (right) shows the growth rate of the inverse link of the logistic loss. When the growth rate is smaller than one, the loss acts as a regularizer of probability estimates. From (13), this regularization multiplies any decrease of prediction error obtained by parameter regularization. This motivates the following definition.

Definition 1 *Let $\phi(v)$ be a proper margin loss. Then*

$$\rho_\phi(v) = \frac{1}{|\{[f_\phi^*]^{-1}\}'(v)|} \tag{14}$$

is the regularization strength of $\phi(v)$. If $\rho_\phi(v) \geq 1, \forall v$, then $\phi(v)$ is denoted a regularization loss.

1. When the optimal link is unique, the loss is denoted strictly proper. Because this is the case for all losses considered in this work, we simply refer to the loss as proper.

3.2 Generalization

An alternative way to characterize the interaction of loss-based and parameter-based regularization is to investigate how the two impact classifier generalization. This can be done by characterizing the dependence of classification error bounds on the two forms of regularization. Since, in this work, we will emphasize boosting algorithms, we rely on the following well known boosting bound.

Theorem 1 *(Schapire et al., 1998) Consider a sample S of m examples $\{(\mathbf{x}_1, y_i), \ldots, (\mathbf{x}_m, y_m)\}$ and a predictor $\hat{p}^*(\mathbf{x}; \mathbf{w})$ of the form of (5) where the $h_i(x)$ are in a space $\mathcal{H}$ of base classifiers of VC-dimension d. Then, with probability at least $1 - \delta$ over the choice of S, for all $\theta > 0$,*

$$P_{\mathbf{X},Y}[yp(\mathbf{x}; \mathbf{w}) \leq 0] \leq P_S \left[\frac{y\hat{p}^*(\mathbf{x}; \mathbf{w})}{||\mathbf{w}||_1} \leq \theta \right] + O\left(\frac{1}{\sqrt{m}} \sqrt{\frac{d \log^2(m/d)}{\theta^2} + \log(1/\delta)} \right),$$

where P_S denotes an empirical probability over the sample S.

Given $\mathcal{H}, m, d$ and δ, the two terms of the bound are functions of θ. The first term depends on the distribution of the margins $y_i\hat{p}^*(\mathbf{x}_i; \mathbf{w})$ over the sample. Assume, for simplicity, that S is separable by $\hat{p}^*(\mathbf{x}; \mathbf{w})$, i.e. $y_i\hat{p}^*(\mathbf{x}_i; \mathbf{w}) > 0, \forall i$, and denote the empirical margin by

$$\gamma_s = y_{i^*}\hat{p}^*(\mathbf{x}_{i^*}; \mathbf{w}), \qquad i^* = \arg\min_i y_i\hat{p}^*(\mathbf{x}_i; \mathbf{w}). \tag{15}$$

Then, for any $\epsilon > 0$ and $\theta = \gamma_s/||\mathbf{w}||_1 - \epsilon$, the empirical probability is zero and

$$P_{\mathbf{X},Y}[yp(\mathbf{x}; \mathbf{w}) \leq 0] \leq O\left(\frac{1}{\sqrt{m}} \sqrt{\frac{d \log^2(m/d)}{(\frac{\gamma_s}{||\mathbf{w}||_1} - \epsilon)^2} + \log(1/\delta)} \right).$$

Using (11) and a first order Taylor series expansion of $[f_\phi^*]^{-1}(.)$ around the origin

$$\begin{aligned} \hat{\eta}(\mathbf{x}_{i^*}) &= [f_\phi^*]^{-1}(y_{i^*}\gamma_s) \\ &\approx [f_\phi^*]^{-1}(0) + y_{i^*}\gamma_s\{[f_\phi^*]^{-1}\}'(0) \end{aligned}$$

it follows that

$$\gamma_s \approx \rho_\phi(0)|\hat{\eta}(\mathbf{x}_{i^*}) - 1/2|, \tag{16}$$

and the bound can be approximated by

$$P_{\mathbf{X},Y}[yp(\mathbf{x}; \mathbf{w}) \leq 0] \leq O\left(\frac{1}{\sqrt{m}} \sqrt{\frac{d \log^2(m/d)}{\left(\frac{\rho_\phi(0)}{||\mathbf{w}||_1}|\hat{\eta}(\mathbf{x}_{i^*}) - 1/2| - \epsilon \right)^2} + \log(1/\delta)} \right). \tag{17}$$

Since this is a monotonically decreasing function of the generalization factor

$$\kappa = \frac{\rho_\phi(0)}{||\mathbf{w}||_1}, \tag{18}$$

larger κ lead to tighter bounds on the probability of classification error, i.e. classifiers with stronger generalization guarantees. This confirms the complimentary nature of parameter and probability regularization, discussed in the previous section. Parameter regularization, as in (10), encourages solutions of smaller $||\mathbf{w}||_1$ and thus larger κ. Regularization losses multiply this effect by the regularization strength $\rho_\phi(0)$. This is in agreement with the multiplicative form of (13). In summary, for regularization losses, the generalization guarantees of classical parameter regularization are amplified by the strength of the probability regularization at the classification boundary.

4. Controlling the Regularization Strength of Proper Losses

In the remainder of this work, we study the design of regularization losses. In particular, we study how to control the regularization strength of a proper loss, by manipulating some loss parameter.

4.1 Proper Losses

The structure of proper losses can be studied by relating conditional risk minimization to the classical problem of probability elicitation in statistics (Savage, 1971; DeGroot and Fienberg, 1983). Here, the goal is to find the probability estimator $\hat{\eta}$ that maximizes the expected score

$$I(\eta, \hat{\eta}) = \eta I_1(\hat{\eta}) + (1 - \eta) I_{-1}(\hat{\eta}), \tag{19}$$

of a scoring rule that assigns to prediction $\hat{\eta}$ a score $I_1(\hat{\eta})$ when event $y = 1$ holds and a score $I_{-1}(\hat{\eta})$ when $y = -1$ holds. The scoring rule is proper if its components $I_1(\cdot), I_{-1}(\cdot)$ are such that the expected score is maximal when $\hat{\eta} = \eta$, i.e.

$$I(\eta, \hat{\eta}) \leq I(\eta, \eta) = J(\eta), \ \ \forall \eta \tag{20}$$

with equality if and only if $\hat{\eta} = \eta$. A set of conditions under which this holds is as follows.

Theorem 2 *(Savage, 1971) Let $I(\eta, \hat{\eta})$ be as defined in (19) and $J(\eta) = I(\eta, \eta)$. Then (20) holds if and only if $J(\eta)$ is convex and*

$$I_1(\eta) = J(\eta) + (1 - \eta) J'(\eta) \qquad I_{-1}(\eta) = J(\eta) - \eta J'(\eta). \tag{21}$$

Several works investigated the connections between probability elicitation and risk minimization (Buja et al., 2006; Masnadi-Shirazi and Vasconcelos, 2008; Reid and Williamson, 2010). We will make extensive use of the following result.

Theorem 3 *(Masnadi-Shirazi and Vasconcelos, 2008) Let $I_1(\cdot)$ and $I_{-1}(\cdot)$ be as in (21), for any continuously differentiable convex $J(\eta)$ such that $J(\eta) = J(1 - \eta)$, and $f(\eta)$ any invertible function such that $f^{-1}(-v) = 1 - f^{-1}(v)$. Then*

$$I_1(\eta) = -\phi(f(\eta)) \qquad I_{-1}(\eta) = -\phi(-f(\eta))$$

if and only if

$$\phi(v) = -J\left(f^{-1}(v)\right) - (1 - f^{-1}(v)) J'\left(f^{-1}(v)\right).$$

It has been shown that, for $C_\phi(\eta, p)$, $f^*_\phi(\eta)$, and $C^*_\phi(\eta)$ as in (7)-(9), $C^*_\phi(\eta)$ is concave (Zhang, 2004) and

$$C^*_\phi(\eta) = C^*_\phi(1-\eta) \quad (22)$$

$$[f^*_\phi]^{-1}(-v) = 1 - [f^*_\phi]^{-1}(v). \quad (23)$$

Hence, the conditions of the theorem are satisfied by any continuously differentiable $J(\eta) = -C^*_\phi(\eta)$ and invertible $f(\eta) = f^*_\phi(\eta)$. It follows that, $I(\eta, \hat{\eta}) = -C_\phi(\eta, f)$ is the expected score of a proper scoring rule if and only if the loss has the form

$$\phi(v) = C^*_\phi\left([f^*_\phi]^{-1}(v)\right) + (1 - [f^*_\phi]^{-1}(v))[C^*_\phi]'\left([f^*_\phi]^{-1}(v)\right). \quad (24)$$

In this case, the predictor of minimum risk is $p^* = f^*_\phi(\eta)$, and posterior probabilities can be recovered with (11). Hence, the loss ϕ is proper and the predictor p^* calibrated. In summary, proper losses have the structure of (22)-(24). In this work, we also assume that $C^*_\phi(0) = C^*_\phi(1) = 0$. This guarantees that the minimum risk is zero when there is absolute certainty about the class label Y, i.e. $P_{Y|\mathbf{X}}(1|\mathbf{x}) = 0$ or $P_{Y|\mathbf{X}}(1|\mathbf{x}) = 1$.

4.2 Loss Margin and Regularization Strength

The facts that 1) the empirical margin γ_s of (15) is a function of the loss margin μ_ϕ of Figure 1, and 2) the regularization strength ρ_ϕ is related to γ_s by (16), suggests that μ_ϕ is a natural loss parameter to control ρ_ϕ. A technical difficulty is that a universal definition of μ_ϕ is not obvious, since most margin losses $\phi(v)$ only converge to zero as $v \to \infty$. Although approximately zero for large positive v, they are strictly positive for all finite v. This is, for example, the case of the logistic loss $\phi(v) = \log(1 + e^{-v})$ of Figure 1 and the boosting loss of Table 1. To avoid this problem, we use a definition based on the second-order Taylor series expansion of ϕ around the origin. The construct is illustrated in Figure 2, where the loss margin μ_ϕ is defined by the point where the quadratic expansion reaches its minimum. It can be easily shown that this is the point $v = \mu_\phi$, where

$$\mu_\phi = -\frac{\phi'(0)}{\phi''(0)}. \quad (25)$$

In Appendix A, we show that, under mild conditions (see Lemma 9) on the inverse link $[f^*_\phi]^{-1}(\eta)$ of a twice differentiable loss ϕ

$$\mu_\phi = \frac{\rho_\phi(0)}{2}, \quad (26)$$

and the regularization strength of ϕ is lower bounded by twice the loss margin

$$\rho_\phi(v) \geq 2\mu_\phi. \quad (27)$$

Under these conditions, $\phi(v)$ is a regularization loss if and only if $\mu_\phi \geq \frac{1}{2}$. This establishes a direct connection between margins and probability regularization: larger loss margins produce more strongly regularized probability estimates. Hence, for proper losses of suitable link, the large margin strategy for classifier learning is also a strategy for regularization of probability estimates. In fact, from (26) and (18), the generalization factor of these losses is directly determined by the loss margin, since $\kappa = \frac{2\mu_\phi}{||\mathbf{w}||_1}$.

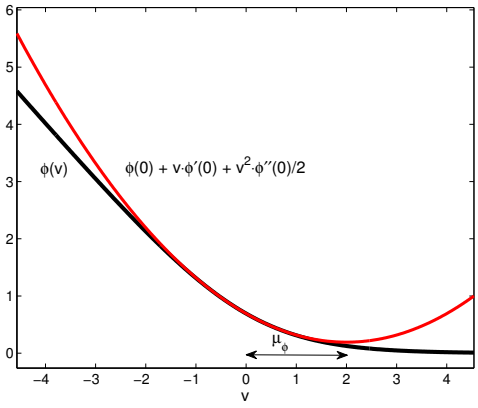


Figure 2: Definition of the loss margin μ_ϕ of a loss ϕ.

4.3 The Generalized Logit Link

As shown in Lemma 9 of Appendix A, the conditions that must be satisfied by the inverse link for (26) and (27) to hold (monotonically increasing, maximum derivative at the origin) are fairly mild. For example, they hold for the scaled logit

$$\gamma(\eta; a) = a \log \frac{\eta}{1-\eta} \qquad \gamma^{-1}(v; a) = \frac{e^{v/a}}{1+e^{v/a}}, \tag{28}$$

which, as shown in Table 1, is the optimal link of the exponential loss when $a = 1/2$ and of the logistic loss when $a = 1$. Since the exponential loss of boosting has margin $\mu_\phi = 1$ and the logistic loss $\mu_\phi = 2$, it follows from the lemma that these are regularization losses. However, the conditions of the lemma hold for many other link functions. In this work, we consider a broad family of such functions, which we denote as the generalized logit.

Definition 2 *An invertible transformation $\pi(\eta)$ is a generalized logit if its inverse, $\pi^{-1}(v)$, has the following properties*

1. *$\pi^{-1}(v)$ is monotonically increasing,*
2. *$\lim_{v\to\infty} \pi^{-1}(v) = 1$*
3. *$\pi^{-1}(-v) = 1 - \pi^{-1}(v)$,*
4. *for finite v, $(\pi^{-1})^{(2)}(v) = 0$ if and only if $v = 0$,*

where $\pi^{(n)}$ is the n^{th} order derivative of π.

In Appendix B, we discuss some properties of the generalized logit and show that all conditions of Lemma 9 hold when $f^*_\phi(\eta)$ is in this family of functions. When combined with Lemma 9, this proves the following result.

Theorem 4 *Let $\phi(v)$ be a twice differentiable proper loss of generalized logit link $f^*_\phi(\eta)$. Then*

$$\mu_\phi = \frac{\rho_\phi(0)}{2} \tag{29}$$

and the regularization strength of $\phi(v)$ is lower bounded by twice the loss margin $\rho_\phi(v) \geq 2\mu_\phi$. $\phi(v)$ is a regularization loss if and only if $\mu_\phi \geq \frac{1}{2}$.

5. Controlling the Regularization Strength

The results above show that it is possible to control the regularization strength of a proper loss of generalized logit link by manipulating the loss margin μ_ϕ. In this section we derive procedures to accomplish this.

5.1 Tunable Regularization Losses

We start by studying the set of proper margin losses whose regularization is controlled by a parameter $\sigma > 0$. These are denoted tunable regularization losses.

Definition 3 *Let $\phi(v)$ be a proper loss of generalized logit link $f^*_\phi(\eta)$. A parametric loss*

$$\phi_\sigma(v) = \phi(v;\sigma) \quad \textit{such that} \quad \phi(v;1) = \phi(v)$$

is the tunable regularization loss generated by $\phi(v)$ if $\phi_\sigma(v)$ is a proper loss of generalized logit link and

$$\mu_{\phi_\sigma} = \sigma\mu_\phi,$$

for all σ such that

$$\sigma \geq \frac{1}{2\mu_\phi}. \tag{30}$$

The parameter σ is the gain of the tunable regularization loss $\phi_\sigma(v)$.

Since, from (29) and (14), the loss margin μ_ϕ only depends on the derivative of the inverse link at the origin, a tunable regularization loss can be generated from any proper loss of generalized logit link, by simple application of Theorem 3.

Lemma 4 *Let $\phi(v)$ be a proper loss of generalized logit link $f^*_\phi(\eta)$. The parametric loss*

$$\phi_\sigma(v) = C^*_{\phi_\sigma}\{[f^*_{\phi_\sigma}]^{-1}(v)\} + (1 - [f^*_{\phi_\sigma}]^{-1}(v))[C^*_{\phi_\sigma}]'([f^*_{\phi_\sigma}]^{-1}(v)), \tag{31}$$

where

$$f^*_{\phi_\sigma}(\eta) \;=\; \sigma f^*_\phi(\eta) \tag{32}$$

$C^*_{\phi_\sigma}(\eta)$ is a minimum risk function (i.e. a continuously differentiable concave function with symmetry $[C^*_{\phi_\sigma}](1-\eta) = [C^*_{\phi_\sigma}](\eta)$) such that $C^*_{\phi_\sigma}(0) = 0$, and (30) holds is a tunable regularization loss.

Proof From (32)

$$[f^*_{\phi_\sigma}]^{-1}(v) = [f^*_\phi]^{-1}\left(\frac{v}{\sigma}\right). \tag{33}$$

Since $[f^*_\phi]^{-1}(v)$ is a generalized logit link it has the properties of Definition 2. Since these properties continue to hold when v is replaced by v/σ, it follows that $f^*_{\phi_\sigma}(v)$ is a generalized logit link. It follows from (31) that $\phi_\sigma(v)$ satisfies the conditions of Theorem 3 and is a proper loss. Since $\mu_{\phi_\sigma} = \frac{\rho_{\phi_\sigma}(0)}{2} = \frac{1}{2\{[f^*_{\phi_\sigma}]^{-1}\}'(0)} = \sigma\mu_\phi$, the parametric loss $\phi_\sigma(v)$ is a tunable regularization loss. ■

In summary, it is possible to generate a tunable regularization loss by simply rescaling the link of a proper loss. Interestingly, this holds independently of how σ parameterizes the minimum risk $[C^*_{\phi_\sigma}](\eta)$. However, not all such losses are useful. If, for example, the process results in

$$\phi_\sigma(v) = \phi(v/\sigma),$$

it corresponds to a simple rescaling of the horizontal axis of Figure 1. The loss $\phi_\sigma(v)$ is thus not fundamentally different from $\phi(v)$. Using this loss in a learning algorithm is equivalent to varying the margin by rescaling the feature space $\mathcal{X}$.

5.2 The Binding Function

To produce non-trivial tunable regularization losses $\phi_\sigma(v)$, we need a better understanding of the role of the minimum risk $[C^*_{\phi_\sigma}](\eta)$. This is determined by the binding function of the loss.

Definition 5 *Let $\phi(v)$ be a proper loss of link $f^*_\phi(\eta)$, and minimum risk $C^*_\phi(\eta)$. The function*

$$\beta_\phi(v) = [C^*_\phi]'\left([f^*_\phi]^{-1}(v)\right) \tag{34}$$

is denoted the binding function of ϕ.

The properties of the binding function are discussed in Appendix C and illustrated in Figure 3. For proper losses of generalized logit link, $\beta_\phi(v)$ is a monotonically decreasing odd function, which determines the behavior of $\phi(v)$ away from the origin and defines a one-to-one mapping between the link f^*_ϕ and the derivative of the risk C^*_ϕ. In this way, β_ϕ "binds" link and risk.

The following result shows that the combination of link and binding function determine the loss up to a constant.

Theorem 5 *Let $\phi(v)$ be a proper loss of generalized logit link $f^*_\phi(\eta)$ and binding function $\beta_\phi(v)$. Then*

$$\phi'(v) \;=\; (1 - [f^*_\phi]^{-1}(v))\beta'_\phi(v). \tag{35}$$

Proof From (24) and the definition of β_ϕ,

$$\phi(v) = C^*_\phi([f^*_\phi]^{-1}(v)) + (1 - [f^*_\phi]^{-1}(v))\beta_\phi(v). \tag{36}$$

Taking derivatives on both sides leads to (35). ■

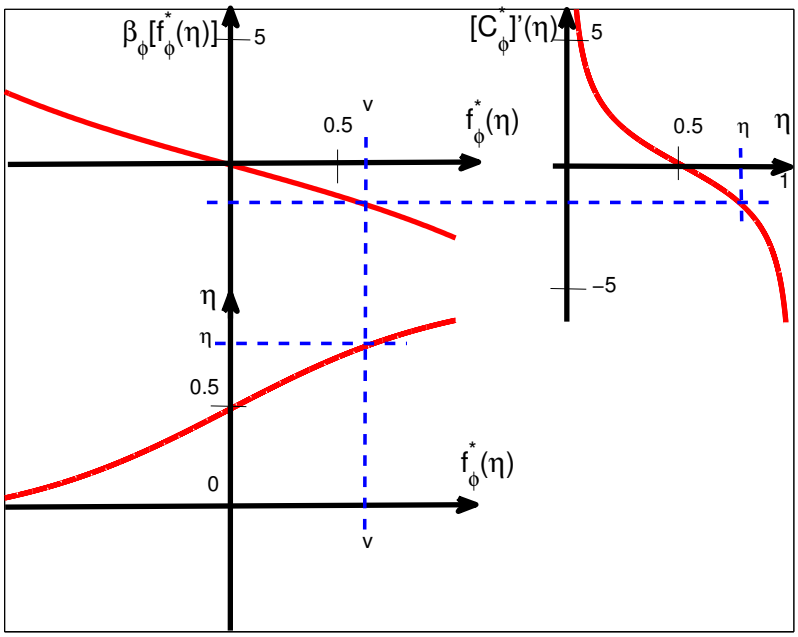


Figure 3: Link $f^*_\phi(\eta)$, risk derivative $[C^*_\phi]'(\eta)$, and binding function $\beta_\phi(f^*_\phi(\eta))$ of a proper loss $\phi(v)$ of generalized logit link.

This result enables the derivation of a number of properties of proper losses of generalized logit link. These are discussed in Appendix D.1, where such losses are shown to be monotonically decreasing, convex under certain conditions on the inverse link and binding function, and identical to the binding function for large negative margins. In summary, a proper loss of generalized logit link can be decomposed into two fundamental quantities: the inverse link, which determines its regularization strength, and the binding function, which determines its behavior away from the origin. Since tunable regularization losses are proper, the combination of this result with Lemma 4 and Definition 5 proves the following theorem.

Theorem 6 *Let $\phi(v)$ be a proper loss of generalized logit link $f^*_\phi(\eta)$. The parametric loss*

$$\phi'_\sigma(v) = (1 - [f^*_{\phi_\sigma}]^{-1}(v))\beta'_{\phi_\sigma}(v), \tag{37}$$

where

$$f^*_{\phi_\sigma}(\eta) \quad = \quad \sigma f^*_\phi(\eta), \tag{38}$$

$\beta_{\phi_\sigma}(v)$ is a binding function (i.e. a continuously differentiable, monotonically decreasing, odd function), and σ is such that (30) holds is a tunable regularization loss.

Algorithm 1: BoostLR

Input: Training set $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$, where $y_i \in \{1, -1\}$ is the class label of example $\mathbf{x}$, regularization gain σ, and number T of weak learners in the final decision rule.

Initialization: Set $G^{(0)}(\mathbf{x}_i) = 0$ and $w^{(1)}(\mathbf{x}_i) = -\left(1 - [f^*_{\phi_\sigma}]^{-1}(y_i G^{(0)}(\mathbf{x}_i))\right) \beta'_{\phi_\sigma}\left(y_i G^{(0)}(\mathbf{x}_i)\right) \quad \forall \mathbf{x}_i$.

for $t = \{1, \ldots, T\}$ **do**

choose weak learner

$$g^*(\mathbf{x}) = \arg\max_{g(\mathbf{x})} \sum_{i=1}^{n} y_i w^{(t)}(\mathbf{x}_i) g(\mathbf{x}_i)$$

update predictor G($\mathbf{x}$)

$$G^{(t)}(\mathbf{x}) = G^{(t-1)}(\mathbf{x}) + g^*(\mathbf{x})$$

update weights

$$w^{(t+1)}(\mathbf{x}_i) = -\left(1 - [f^*_{\phi_\sigma}]^{-1}(y_i G^{(t)}(\mathbf{x}_i))\right) \beta'_{\phi_\sigma}\left(y_i G^{(t)}(\mathbf{x}_i)\right) \quad \forall \mathbf{x}_i$$

end for

Output: decision rule $h(\mathbf{x}) = \text{sgn}[G^{(T)}(\mathbf{x})]$.

5.3 Boosting With Tunable Probability Regularization

Given a tunable regularization loss ϕ_σ, various algorithms can be used to design a classifier. Boosting accomplishes this by gradient descent in a space $\mathcal{W}$ of weak learners. While there are many variants, in this work we adopt the GradientBoost framework (Friedman, 2001). This searches for the predictor $G(\mathbf{x})$ of minimum empirical risk on a sample $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$,

$$R(G) = \sum_{i=1}^{n} \phi_\sigma(y_i G(\mathbf{x}_i)).$$

At iteration t, the predictor is updated according to

$$G^{(t)}(\mathbf{x}) = G^{(t-1)}(\mathbf{x}) + g^{(t)}(\mathbf{x}), \tag{39}$$

where $g^{(t)}(\mathbf{x})$ is the gradient of $R(G)$ in $\mathcal{W}$, i.e. the weak learner

$$\begin{aligned} g^{(t)}(\mathbf{x}) &= \arg\max_{g} \sum_{i=1}^{n} -y_i \phi'_\sigma(y_i G^{(t-1)}(\mathbf{x}_i)) g(\mathbf{x}_i) \\ &= \arg\max_{g} \sum_{i=1}^{n} y_i w^{(t)}_\sigma(\mathbf{x}_i) g(\mathbf{x}_i), \end{aligned}$$

where

$$w^{(t)}_\sigma(\mathbf{x}_i) = -\phi'_\sigma(y_i G^{(t-1)}(\mathbf{x}_i))$$

is the weight of example $\mathbf{x}_i$ at iteration t. For a tunable regularization loss $\phi_\sigma(v)$ of generalized logit link $f^*_{\phi_\sigma}(\eta)$ and binding function $\beta_{\phi_\sigma}(v)$, it follows from (37) that

$$w^{(t)}_\sigma(\mathbf{x}_i) = -\left(1 - [f^*_{\phi_\sigma}]^{-1}\left(y_i G^{(t-1)}(\mathbf{x}_i)\right)\right)\beta'_{\phi_\sigma}\left(y_i G^{(t-1)}(\mathbf{x}_i)\right). \tag{40}$$

Boosting with these weights is denoted boosting with loss regularization (BoostLR) and summarized in Algorithm 1.

The weighting mechanism of BoostLR provides some insight on how the choices of link and binding function affect classifier behavior. Using $\gamma_i = y_i G^{(t-1)}(\mathbf{x}_i)$ to denote the margin of $\mathbf{x}_i$ for the classifier of iteration $t-1$,

$$w^{(t)}_\sigma(\mathbf{x}_i) = -\phi'_\sigma(\gamma_i) = -\left(1 - [f^*_{\phi_\sigma}]^{-1}(\gamma_i)\right)\beta'_{\phi_\sigma}(\gamma_i). \tag{41}$$

It follows from the discussion of the previous section that 1) the link $f^*_{\phi_\sigma}$ is responsible for the behavior of the weights around the classification boundary and 2) the binding function β_{ϕ_σ} for the behavior at large margins. For example, applying (34) to the links and risks of Table 1 results in

$$\beta(v) = e^{-v} - e^v \qquad \beta'(v) = -e^{-v} - e^v \tag{42}$$

for AdaBoost and

$$\beta(v) = -v \qquad \beta'(v) = -1 \tag{43}$$

for LogitBoost. In result, AdaBoost weights are exponentially large for examples of large negative margin γ_i, while LogitBoost weights remain constant. This fact has been used to explain the much larger sensitivity of AdaBoost to outliers (Maclin and Opitz, 1997; Dietterich, 2000; Mason et al., 2000; Masnadi-Shirazi and Vasconcelos, 2008; Friedman et al., 2000; McDonald et al., 2003; Leistner et al., 2009). Under this view, the robustness of a boosting algorithm to outliers is determined by its binding function. Hence, the decomposition of a loss into link and binding functions translates into a functional decomposition for boosting algorithms. It decouples the generalization ability of the learned classifier, determined by the regularization strength imposed by the link, from its robustness to outliers, determined by the binding function.

6. The Set of Tunable Regularization Losses

The link-binding decomposition can also be used to characterize the structure of the set of tunable regularization losses.

6.1 Equivalence Classes

A simple consequence of (37) is that the set $\mathcal{R}$ of tunable regularization losses ϕ_σ is the Cartesian product of the set $\mathcal{L}$ of generalized logit links and the set $\mathcal{B}$ of binding functions. It follows that both generalized logit links f_σ and binding functions β_σ define equivalence classes in $\mathcal{R}$. In fact, $\mathcal{R}$ can be partitioned according to

$$\mathcal{R} = \cup_{\beta_\sigma}\mathcal{R}_{\beta_\sigma} \quad \text{where} \quad \mathcal{R}_{\beta_\sigma} = \{\phi_\sigma | \beta_{\phi_\sigma} = \beta_\sigma\}$$

or

$$\mathcal{R} = \cup_{f_\sigma}\mathcal{R}_{f_\sigma} \quad \text{where} \quad \mathcal{R}_{f_\sigma} = \{\phi_\sigma | f^*_{\phi_\sigma} = f_\sigma\}.$$

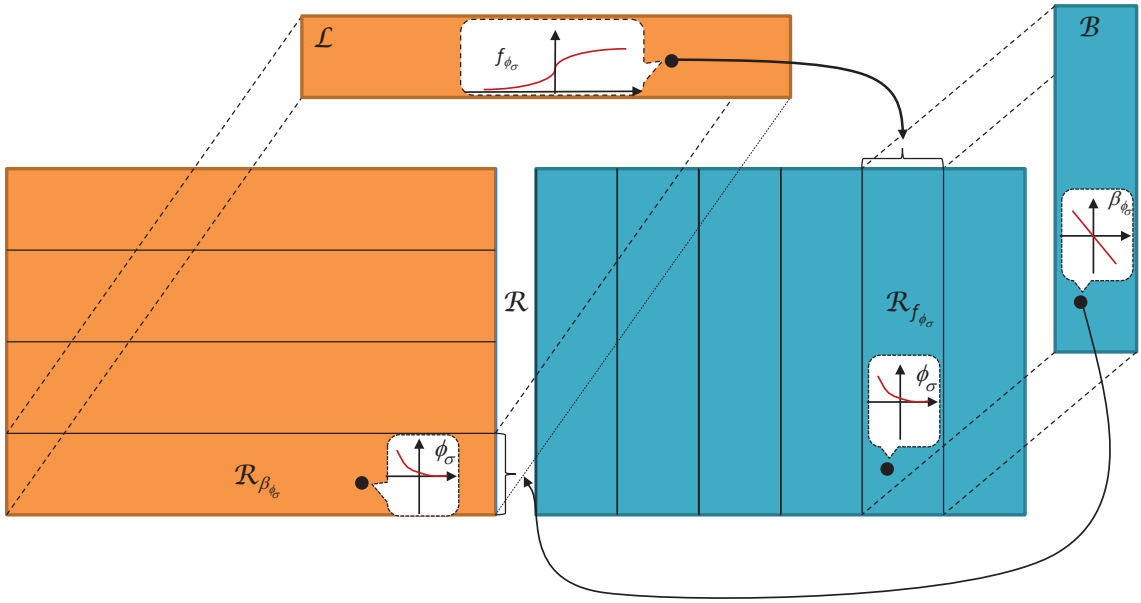


Figure 4: The set $\mathcal{R}$ of tunable regularization losses can be partitioned into equivalence classes $\mathcal{R}_{f_{\phi\sigma}}$, isometric to the set $\mathcal{B}$ of binding functions, or equivalence classes $\mathcal{R}_{\beta_{\phi\sigma}}$, isometric to the set $\mathcal{L}$ of generalized logit links. A tunable regularization loss ϕ_σ is defined by a pair of link f_{ϕ_σ} and binding β_{ϕ_σ} functions.

The sets $\mathcal{R}_{f_\sigma}$ are isomorphic to $\mathcal{B}$, which is itself isomorphic to the set of continuously differentiable, monotonically decreasing, odd functions. The sets $\mathcal{R}_{\beta_\sigma}$ are isomorphic to $\mathcal{L}$, which is shown to be isomorphic, in Appendix B.2, to the set of parametric continuous scale probability density functions (pdfs)

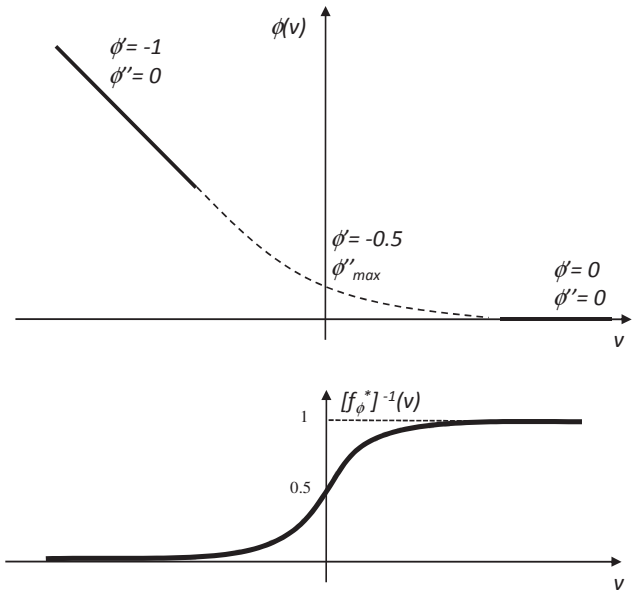
$$\psi_\sigma(v) = \frac{1}{\sigma}\psi\left(\frac{v}{\sigma}\right), \tag{44}$$

where $\psi(v)$ has unit scale, a unique maximum at the origin, and $\psi(-v) = \psi(v)$. The structure of the set of tunable regularization losses is illustrated in Figure 4. The set can be partitioned in two ways. The first is into a set of equivalence classes $\mathcal{R}_{\beta_\sigma}$ isomorphic to the set of pdfs of (44). The second into a set of equivalence classes $\mathcal{R}_{f_\sigma}$ isomorphic to the set of monotonically decreasing odd functions.

6.2 Design of Regularization Losses

An immediate consequence of the structure of $\mathcal{R}$ is that all tunable regularization losses can be designed by the following procedure.

1. select a scale pdf $\psi_\sigma(v)$ with the properties of (44).
2. set $[f^*_{\phi_\sigma}]^{-1}(v) = c_\sigma(v)$, where $c_\sigma(v) = \int_{-\infty}^{v} \psi_\sigma(q)dq$ is the cumulative distribution function (cdf) of $\psi_\sigma(v)$.



Given ψ_σ or c_σ
$\phi_\sigma = \int [c_\sigma(v) - 1] dv$
$[f^*_{\phi_\sigma}]^{-1}(v) = c_\sigma(v)$
$C^*_{\phi_\sigma} = -\int_0^\eta c_\sigma^{-1}(\nu) d\nu$

Given ϕ_σ, $f^*_{\phi_\sigma}$ or $C^*_{\phi_\sigma}$
$\psi_\sigma(v) = \phi''_\sigma(v)$
$\psi_\sigma(v) = \{[f^*_{\phi_\sigma}]^{-1}\}'(v)$
$c_\sigma(v) = [-\{C^*_{\phi_\sigma}\}']^{-1}(v)$

Figure 5: Canonical regularization losses. Left: general properties of the loss and inverse link functions. Right: Relations between losses and scale pdfs.

3. select a binding function $\beta_{\phi_\sigma}(v)$. This can be any parametric family of continuously differentiable, monotonically decreasing, odd functions.

4. define the tunable regularization loss as $\phi'_\sigma(v) = (1 - [f^*_{\phi_\sigma}]^{-1}(v))\beta'_{\phi_\sigma}(v)$.

5. restrict σ according to (30).

Note that the derivative $\phi'_\sigma(v)$ is sufficient to implement the BoostLR algorithm. If desired, it can be integrated to produce a formula for the loss $\phi_\sigma(v)$. This defines the loss up to a constant, which can be determined by imposing the constraint that $\lim_{v\to\infty} \phi_\sigma(v) = 0$. As discussed in the previous section, this procedure enables the independent control of the regularization strength and robustness of the losses $\phi_\sigma(v)$. In fact, it follows from step 2. and (14) that

$$\rho_{\phi_\sigma}(v) = \frac{1}{\psi_\sigma(v)}, \tag{45}$$

i.e. the choice of pdf $\psi_\sigma(v)$ determines the regularization strength of $\phi_\sigma(v)$. The choice of binding function in step 3. then limits $\phi_\sigma(v)$ to an equivalence class $\mathcal{R}_{\beta_\sigma}$ of regularization losses with common robustness properties. We next consider some important equivalence classes.

6.3 Canonical Regularization Losses

We start by considering the set of tunable regularization losses with linear binding function

$$\beta_{\phi_\sigma}(v) = -v. \tag{46}$$

	Generalized Logistic (GLog)	Generalized Gaussian (GGauss)
$\psi_\sigma(v)$	$\frac{e^{\frac{v}{\sigma}}}{\sigma(1+e^{\frac{v}{\sigma}})^2}$	$\frac{1}{4\sigma}e^{-(\frac{\sqrt{\pi}}{4\sigma}v)^2}$
$c_\sigma(v)$	$\frac{e^{v/\sigma}}{1+e^{v/\sigma}}$	$\frac{1}{2}\left[1+erf\left(\frac{\sqrt{\pi}}{4\sigma}v\right)\right]$
$\phi_\sigma(v)$	$\sigma\log\left(1+e^{-\frac{v}{\sigma}}\right)$	$\frac{v}{2}\left[erf\left(\frac{\sqrt{\pi}}{4\sigma}v\right)-1\right]+\frac{2\sigma}{\pi}e^{-\left(\frac{\sqrt{\pi}}{4\sigma}v\right)^2}$
$f^*_{\phi_\sigma}(\eta)$	$\sigma\log\frac{\eta}{1-\eta}$	$\frac{4\sigma}{\sqrt{\pi}}\cdot erf^{-1}(2\eta-1)$
$C^*_{\phi_\sigma}(\eta)$	$-\sigma\eta\log(\eta)-\sigma(1-\eta)\log(1-\eta)$	$-\frac{4\sigma}{\sqrt{\pi}}\int erf^{-1}(2\eta-1)d\eta$
$\rho_\phi(v)$	$\frac{\sigma(1+e^{\frac{v}{\sigma}})^2}{e^{\frac{v}{\sigma}}}$	$4\sigma e^{(\frac{\sqrt{\pi}}{4\sigma}v)^2}$
	Generalized Laplacian (GLaplacian)	**Generalized Boosting (GBoost)**
$\psi_\sigma(v)$	$\frac{1}{4\sigma}e^{-\frac{\lvert v\rvert}{2\sigma}}$	$\frac{2}{\sigma\left(4+(\frac{v}{\sigma})^2\right)^{\frac{3}{2}}}$
$c_\sigma(v)$	$\frac{1}{2}\left[1+sign(v)\left(1-e^{-\frac{\lvert v\rvert}{2\sigma}}\right)\right]$	$\frac{1}{2}+\frac{\frac{v}{\sigma}}{2\sqrt{4+\left(\frac{v}{\sigma}\right)^2}}$
$\phi_\sigma(v)$	$\sigma e^{\frac{-\lvert v\rvert}{2\sigma}}+\frac{1}{2}(\lvert v\rvert-v)$	$\frac{\sigma}{2}\left(\sqrt{4+\left(\frac{v}{\sigma}\right)^2}-\frac{v}{\sigma}\right)$
$f^*_{\phi_\sigma}(\eta)$	$-2\sigma sign(2\eta-1)\log(1-\lvert 2\eta-1\rvert)$	$\sigma\frac{2\eta-1}{\sqrt{\eta(1-\eta)}}$
$C^*_{\phi_\sigma}(\eta)$	$\sigma(1-\lvert 2\eta-1\rvert)[1-\log(1-\lvert 2\eta-1\rvert)]$	$2\sigma\sqrt{\eta(1-\eta)}$
$\rho_\phi(v)$	$4\sigma e^{\frac{\lvert v\rvert}{2\sigma}}$	$\frac{\sigma}{2}\left(4+(\frac{v}{\sigma})^2\right)^{\frac{3}{2}}$

Table 2: Canonical tunable regularization losses

From (37), these losses are uniquely determined by their link function

$$\phi'_\sigma(v) = -(1-[f^*_{\phi_\sigma}]^{-1}(v)). \tag{47}$$

Their properties are discussed in Appendix D.2. As illustrated in Figure 5, they are convex, monotonically decreasing, linear (with slope -1) for large negative v, constant for large positive v, and have slope $-.5$ and maximum curvature at the origin. The only degrees of freedom are in the vicinity of the origin, and determine the loss margin, since $\mu_{\phi_\sigma} = \frac{1}{2\phi''_\sigma(0)}$. Furthermore, because these losses have regularization strength $\rho_{\phi_\sigma}(0) = \frac{1}{\phi''_\sigma(0)}$, they are direct regularizers of probability scores, and regularization losses whenever $\phi''_\sigma(0) \leq 1$. This is reminiscent of a well known result (Bartlett et al., 2006) that Bayes consistency holds for a convex $\phi(v)$ if and only if $\phi'(0) \leq 0$. From Property 4. of Lemma 13, this holds for all regularization losses with the form of (47). The constraint $\phi''_\sigma(0) \leq 1$ is also equivalent to $\frac{\phi''(0)}{\sigma} \leq 1$. This is the condition of (30) for the losses of (47).

When (46) holds, it follows from (34) that $f^*_\phi(\eta) = -[C^*_\phi]'(\eta)$. Buja et al. showed that the empirical risk of (4) is convex when ϕ is a proper loss and this relationship holds . They denoted as canonical risks the risks of (7) for which this is the case (Buja et al., 2006). For consistency, we denote the associated $\phi(v)$ a canonical loss. This is summarized by the following definition.

Definition 6 *A tunable regularization loss $\phi_\sigma(v)$ such that (47) holds for any σ such that $\phi''_\sigma(0) \leq 1$ is a canonical loss.*

We note, however, that what makes canonical losses special is not the guarantee of a convex risk, but that they have the simplest binding function with this guarantee. From Property 2. of Lemma 13, loss convexity does not require a linear binding function. On the other hand, since 1) any risk of convex loss is convex, 2) (57) holds for the linear binding function, and 3) binding functions are monotonically decreasing, the linear binding function is the simplest that guarantees a convex risk.

It should also be noted that the equivalence class of (46) includes many regularization losses. The relations of Figure 5, where $c_\sigma(v)$ is the cumulative distribution function (cdf) of the pdf $\psi_\sigma(v)$ of (44), can be used to derive losses from pdfs or pdfs from losses. Some example tunable canonical regularization losses are presented in Table 2. The generalized logistic (GLog), Gaussian (GGauss), and Laplacian (GLaplacian) losses are tunable losses derived from the logistic, Gaussian, and Laplace pdfs respectively. The GBoost loss illustrates some interesting alternative possibilities for this loss design procedure. In this case, we did not start from the pdf $\psi_\sigma(v)$ but from the minimum risk of boosting (see Table 1). We then used the top equations of Figure 5 to derive the cdf $c_\sigma(v)$ and the bottom equations to obtain $\phi_\sigma(v)$ and $f^*_{\phi_\sigma}(\eta)$. The resulting pdf $\psi_\sigma(v)$ is a special case of the Pearson type VII distribution with zero location parameter, shape parameter $\frac{3}{2}$ and scale parameter 2σ. These losses, their optimal inverse links, and regularization strength are plotted in Figure 6, which also shows how the regularization gain σ influences the loss around the origin, both in terms of its margin properties and regularization strength. Note that, due to (45), canonical losses implement all regularization behaviors possible for tunable regularization losses. This again justifies the denomination of "canonical regularization losses," although such an interpretation does not appear to have been intended by Buja et al.

The combination of BoostLR with a canonical loss is denoted a canonical BoostLR algorithm. For a proper loss ϕ_σ, $G^{(t)}(\mathbf{x})$ converges asymptotically to the optimal predictor $p^*_\sigma(\mathbf{x}) = f^*_{\phi_\sigma}(\eta(\mathbf{x}))$ and the weight function of (40) to

$$w^*(\mathbf{x}_i) \;=\; \begin{cases} 1-\eta(\mathbf{x}_i) & \text{if } y_i = 1 \\ \eta(\mathbf{x}_i) & \text{if } y_i = -1. \end{cases}$$

Hence, the weights of canonical BoostLR converge to the posterior example probabilities. Figure 7 shows the weight functions of the losses of Table 2. An increase in regularization gain σ simultaneously 1) extends the region of non-zero weight away from the boundary, and 2) reduces the derivative amplitude, increasing regularization strength. Hence, larger gains increase both the classification margin and the regularization of probability estimates.

6.4 Shrinkage Losses

Definition 7 *A tunable regularization loss $\phi_\sigma(v)$ such that*

$$\beta'_{\phi_\sigma}(v) = \beta'_\phi\left(\frac{v}{\sigma}\right), \tag{48}$$

for some $\beta_\phi(v) \in \mathcal{B}$ is a shrinkage loss.

Note that, since (48) holds for the linear binding function of (46), canonical regularization losses are shrinkage losses. These losses are easily identifiable, since combining (48), (37),

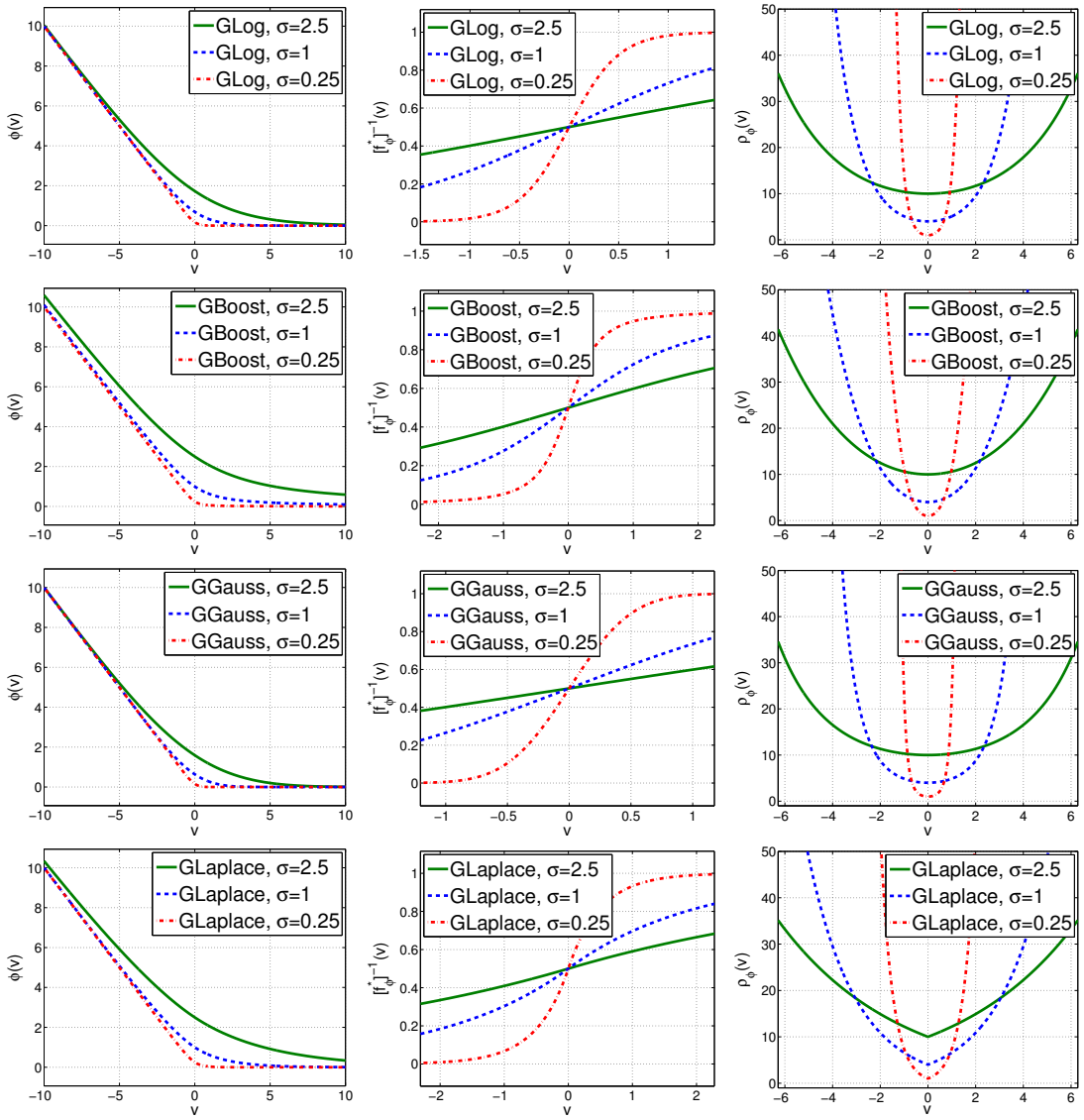


Figure 6: Loss (left), inverse link (middle), and regularization strength (right) functions, for various canonical regularization losses and gains σ. From top to bottom: GLog, GBoost, GGauss and GLaplacian.

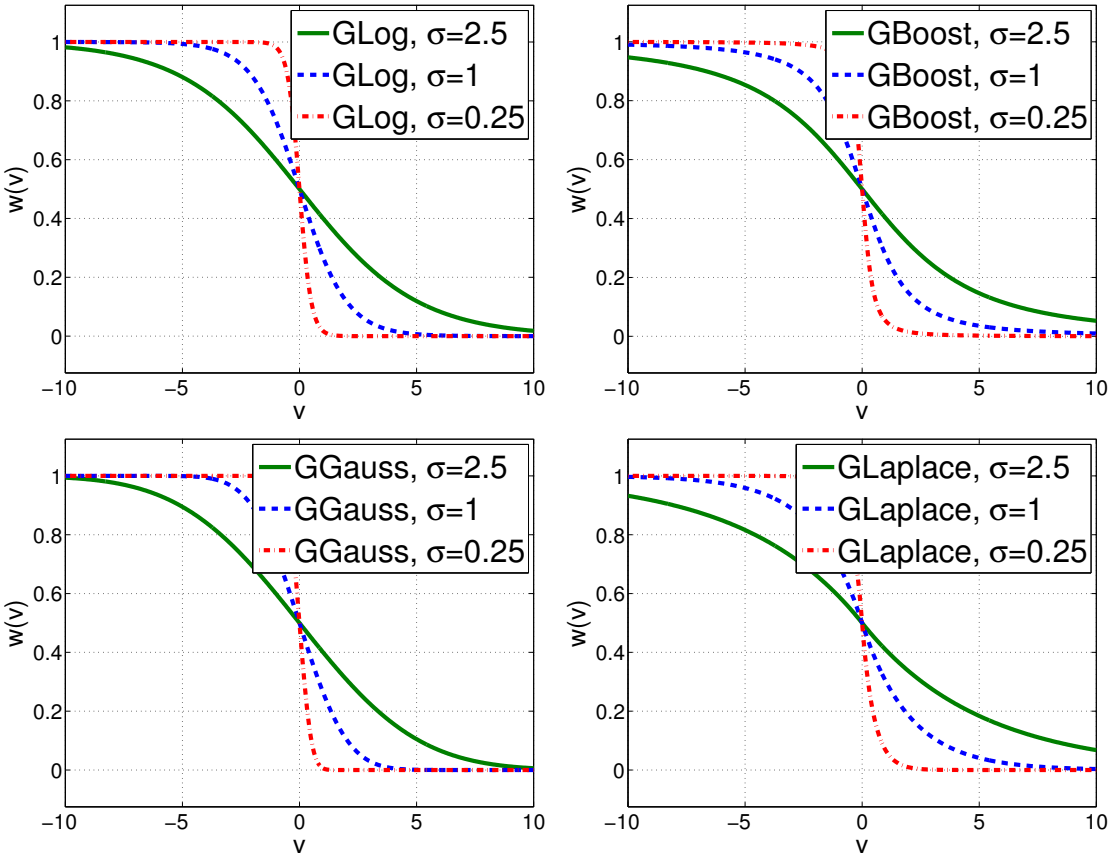


Figure 7: BoostLR weights for various parametric regularization losses and gains. GLog (top left), GBoost (top right), GGauss (bottom left) and GLaplace (bottom right).

and (33) leads to $\phi'_\sigma(v) = \phi'(v/\sigma)$. Hence, ϕ_σ is a shrinkage loss if and only if

$$\phi_\sigma(v) = \sigma\phi\left(\frac{v}{\sigma}\right). \tag{49}$$

This enables the generalization of any proper loss of generalized logit link into a shrinkage loss. For example, using Table 1, it is possible to derive the shrinkage losses generated by the logistic

$$\phi_\sigma(v) = \sigma \log(1 + e^{-\frac{v}{\sigma}})$$

and the exponential loss

$$\phi_\sigma(v) = \sigma e^{-\frac{v}{\sigma}}.$$

The former is the GLog loss of Table 2, but the later is not a canonical regularization loss.

Shrinkage losses also connect BoostLR to shrinkage, a popular regularization heuristic (Hastie et al., 2001). For GradientBoost, this consists of modifying the learning rule

of (39) into

$$G^{(t)}(\mathbf{x}) = G^{(t-1)}(\mathbf{x}) + \lambda g^{(t)}(\mathbf{x}), \tag{50}$$

where $0 < \lambda < 1$ is a learning rate. Shrinkage is inspired by parameter regularization methods from the least-squares regression literature, where similar modifications follow from the adoption of Bayesian models with priors that encourage sparse regression coefficients. This interpretation does not extend to classification, barring the assumption of the least-squares loss and some approximations (Hastie et al., 2001). In any case, it has been repeatedly shown that small learning rates ($\lambda \leq 0.1$) can significantly improve the generalization ability of the learned classifiers. Hence, despite its tenuous theoretical justification, shrinkage is a commonly used regularization procedure.

Shrinkage losses, and the proposed view of margin losses as regularizers of probability estimates, provide a much simpler and more principled justification for the shrinkage procedure. It suffices to note that the combination of (49) and (41) leads to

$$\begin{aligned} w_\sigma^{(t)}(\mathbf{x}_i) &= -\phi'_\sigma(\gamma_i) = -\phi'\left(\frac{\gamma_i}{\sigma}\right) \\ &= -\left(1 - [f^*_\phi]^{-1}\left(\frac{\gamma_i}{\sigma}\right)\right)\beta'_\phi\left(\frac{\gamma_i}{\sigma}\right), \end{aligned}$$

where $\gamma_i = y_i G^{(t-1)}(\mathbf{x}_i)$. Letting $\lambda = 1/\sigma$, this is equivalent to

$$w_\lambda(\mathbf{x}_i) = -\left(1 - [f^*_\phi]^{-1}\left(y_i \lambda G(\mathbf{x}_i)\right)\right)\beta'_\phi\left(y_i \lambda G(\mathbf{x}_i)\right).$$

Hence, the weight function of BoostLR with shrinkage loss ϕ_σ and predictor $G(\mathbf{x})$ is equivalent to the weight function of standard GradientBoost with loss ϕ and shrinked predictor $1/\sigma G(\mathbf{x})$. Since the only other effect of replacing (39) with (50) is to rescale the final predictor $G^{(T)}(\mathbf{x})$, the decision rule $h(\mathbf{x})$ produced by the two algorithms is identical. In summary, GradientBoost with shrinkage and a small learning rate λ is equivalent to BoostLR with a shrinkage loss of large regularization strength $(1/\lambda)$. This justifies the denomination of "shrinkage losses" for the class of regularization losses with the property of (48).

It should be noted, however, that while rescaling the predictor does not affect the decision rule, it affects the recovery of posterior probabilities from the shrinked predictor. The regularization view of shrinkage makes it clear that the probabilities can be recovered with

$$\hat{\eta}(\mathbf{x}) = [f^*_{\phi_\sigma}]^{-1}\left(G^{(T)}(\mathbf{x})\right) = [f^*_\phi]^{-1}\left(\lambda G^{(T)}(\mathbf{x})\right). \tag{51}$$

In the absence of this view, it is not obvious why shrinkage, which is justified as a simple change of learning rate, would require a modified link function for probability recovery. It is also neither clear nor it has been claimed that shrinkage would improve the quality of probability estimates. On the other hand, the discussion above suggests that this is why it works: shrinkage is a procedure for controlling probability regularization strength by manipulation of the loss margin. In fact, since GradientBoost with shrinkage and a small learning rate λ is equivalent to BoostLR with a shrinkage loss of large regularization strength $(1/\lambda)$, Section 3.2 provides a theoretical justification for the empirical evidence that shrinkage improves generalization performance.

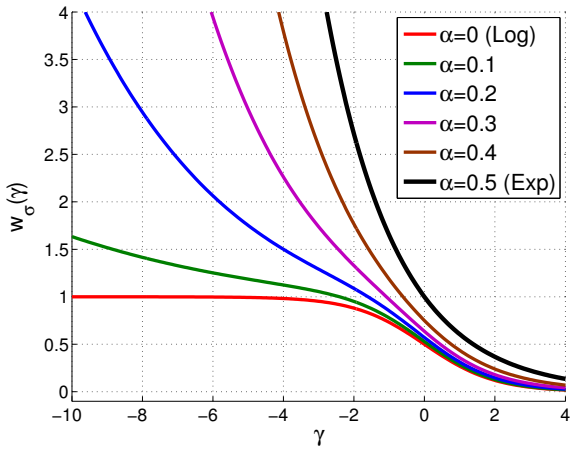


Figure 8: Weight function of the α-tunable regularization loss, for different values of α.

6.5 α-tunable Regularization Losses

From (48), the key to the equivalence between loss-based regularization and shrinkage is the identical parameterization of $[f^*_{\phi_\sigma}]^{-1}(v)$ and $\beta'_{\phi_\sigma}(v)$ in (33) and (48). When this is not the case, BoostLR weights are given by

$$\begin{aligned} w_\sigma(\mathbf{x}_i) &= -\left(1 - [f^*_{\phi_\sigma}]^{-1}(\gamma_i)\right)\beta'_{\phi_\sigma}(\gamma_i) \\ &= -\left(1 - [f^*_{\phi}]^{-1}(\lambda\gamma_i)\right)\beta'_{\phi_\sigma}(\gamma_i) \\ &\neq -\left(1 - [f^*_{\phi}]^{-1}(\lambda\gamma_i)\right)\beta'_{\phi}(\lambda\gamma_i)), \end{aligned}$$

and the shrinkage interpretation no longer holds. One such loss class is defined as follows.

Definition 8 *A tunable regularization loss $\phi_\sigma(v)$ such that*

$$\beta'_{\phi_\sigma}(v) = g(\alpha)\beta'_\phi\left(\alpha\frac{v}{\sigma}\right),$$

where $\beta_\phi(v) \in \mathcal{B}$, $g(\alpha)$ is a constant that depends on α, and $\alpha \geq 0$ is denoted α-tunable.

The additional α parameter enables α-tunable losses to independently control the link and binding functions. In fact, they generalize the previous two loss classes, reducing to shrinkage losses when $\alpha = 1$ and $g(1) = 1$ and canonical losses when $\alpha = 0$ and $g(0)\beta'_\phi(0) = 1$. More generally, the α parameter allows the "interpolation" between pairs of canonical or shrinkage losses of equal generalized logit link. For example, the logistic and exponential losses have the scaled logit of (28) as link function, with $a = 1$ and $a = \frac{1}{2}$, respectively. Since these can be written as $a = \frac{1}{\xi+1}$, for $\xi = 0$ and $\xi = 1$, scaled logits with $\xi \in [0, 1]$ interpolate between the links of the two losses. Similarly, the binding functions of the two losses, given by (42) and (43), are special cases of

$$\beta'_\phi(v) = -\frac{1}{2-b}(e^{-bv} + e^{bv}) \tag{52}$$

with $b = 0$ and $b = 1$. Hence, binding functions with $b = \xi$ and $\xi \in [0, 1]$ interpolate between the binding functions of the two losses. It follows that

$$\phi'(v) = -\left(1 - \frac{e^{(\xi+1)v}}{1 + e^{(\xi+1)v}}\right) \frac{1}{2-\xi}(e^{-\xi v} + e^{\xi v}), \qquad \xi \in [0, 1]$$

interpolates between the derivative of the logistic ($\xi = 0$) and exponential ($\xi = 1$) losses. The derivative of the tunable regularization loss that it generates is

$$\phi'_\mu(v) = -\left(1 - \frac{e^{(\xi+1)\frac{v}{\mu}}}{1 + e^{(\xi+1)\frac{v}{\mu}}}\right) \frac{1}{2-\xi}(e^{-\xi\frac{v}{\mu}} + e^{\xi\frac{v}{\mu}}), \qquad \xi \in [0, 1].$$

Defining $\sigma = \frac{\mu}{\xi+1}$ and $\alpha = \frac{\xi}{1+\xi}$, this can be written as

$$\phi'_\sigma(v) = -\left(1 - \frac{e^{\frac{v}{\sigma}}}{1 + e^{\frac{v}{\sigma}}}\right) \frac{1-\alpha}{2-3\alpha}(e^{-\alpha\frac{v}{\sigma}} + e^{\alpha\frac{v}{\sigma}}), \qquad \alpha \in \left[0, \frac{1}{2}\right], \tag{53}$$

i.e. a α-tunable loss of scaled logit link, $g(\alpha) = \frac{1-\alpha}{2-3\alpha}$, and the binding function of (52). Figure 8 shows the weight function, $w_\sigma(\gamma) = -\phi'_\sigma(\gamma)$, of this loss as a function of the normalized margin $\gamma = v/\sigma$, for different values of α. As α varies, the weight function interpolates between the asymptotically constant weights of LogitBoost (less outlier sensitivity) and the exponential weights of AdaBoost (more sensitive to outliers).

Note that, due to their ability to independently control the link and binding functions, α-tunable losses can always implement this type of interpolation. This can be used to design losses that adapt to the presence of outliers in the data, by cross-validation of α. It should be noted, however, that not all values of $\alpha \geq 0$ lead to sensible loss functions. This is due to the fact that (49) does not hold for these losses. For shrinkage losses, where the property holds, $\phi_\sigma(v) \to 0$ as $v \to \infty$ (whenever $\phi(v)$ has this property), guaranteeing that examples of large positive margin have zero weight. For α-tunable losses, where (49) does not hold, $\beta'_{\phi_\sigma}(v)$ can decrease to $-\infty$ faster than $1 - [f^*_{\phi_\sigma}]^{-1}(v)$ goes to zero, as $v \to \infty$. In this case, examples of large positive margin can receive large positive weight, which is usually undesirable. The losses of (53) have this behavior for $\alpha > 1/2$.

7. Experiments

In this section we discuss various experiments conducted to evaluate different properties of probability regularization.

7.1 Experiments on Two Gaussian Classes

To gain some insight on probability regularization, we considered a simple classification problem, composed of two Gaussian classes of identity covariance, $\mathbf{\Sigma} = \mathbf{I}$, on a two-dimensional space. The means were set to $(0, 0)$ and $(0.7416, 0.7416)$, so as to produce a problem with a Bayes error of 30%. Classifiers were learned with training sets of variable size and evaluated with a test set of $10,000$ examples. All classifiers were learned with BoostLR and the GLog loss, using histogram-based weak learners (Masnadi-Shirazi and

Vasconcelos, 2011; Rasolzadeh et al., 2006; Wu et al., 2004). We started by investigating how the probability estimates varied with the regularization gain σ. The accuracy of the probability estimates was measured by the mean squared error

$$MSE = \frac{1}{n}\sum_{i-1}^{n}[\eta(\mathbf{x}_i) - \hat{\eta}(\mathbf{x}_i)]^2, \tag{54}$$

where $\eta(\mathbf{x}_i)$ and $\hat{\eta}(\mathbf{x}_i)$ are the true and estimated posterior probability for test example $\mathbf{x}_i$. The latter was obtained with (51), where $G^{(T)}(\mathbf{x})$ is the predictor learned by BoostLR. Three regimes were considered. The very small sample regime, where the training set contained $N = 5$ examples per class, the moderate sample size regime, where $N = 40$ and the large sample regime, where $N = 1,000$. Classifiers were learned with BoostLR under the three regimes, for a range of values of σ in the interval $[0.5, 1000]$. Figure 9 shows two complementary views of the MSE data. The top row presents the classical curves of MSE vs. number of boosting iterations T, for different regularization gains. These plots are most useful to assess overfitting, which happens when there is a range of T over which the MSE increases. It is clear that, for both the small and moderate sample sizes, all classifiers eventually overfit as the number of boosting iterations increases, while no overfitting is observed for large sample sizes. The bottom row is most useful to assess the impact of predictor regularization. The data is the same, but these plots show the evolution of the MSE with σ for fixed T. In this case, overfitting occurs on the left of each plot (small values of σ, not enough regularization) and underfitting (too much regularization) on the right.

Overall, the plots demonstrate the complementarity between loss-based probability regularization and classic parameter regularization (due to early stopping, i.e. limiting the number of weak learners in the final ensemble). This is most clear in the moderate sample regime, where many of the curves of the middle column of Figure 9 (top) have the same minimum. Varying the gain σ shifts this minimum, i.e. makes it occur at different numbers of boosting iterations. Hence, when a regularization loss is used, there is less need for early stopping (parameter regularization). This explains the empirical observation that boosted classifiers can do well even with little parameter regularization (e.g. boosted object detectors with thousands of weak learners commonly used in computer vision (Viola and Jones, 2004)). The problem with early stopping is that it can be insufficient for small samples. This is visible in the left column of Figure 9 (top), where there is too little data and boosting overfits *even in the earliest iterations*. The same happens for the moderate sample size (middle column of Figure 9 top) when the regularization gain is small. In these cases, by amplifying parameter based regularization, loss-based regularization can substantially improve the quality of probability estimates. For example, larger σ lead to significant gains in estimation accuracy, for all numbers of boosting iterations, in the left column of Figure 9 (bottom). As σ increases, the best early-stopped MSE ($T = 2$) decreases from roughly 20% to about 5%. Hence, for small samples, loss-based regularization is much more effective than early stopping.

In summary, loss-based regularization is a more flexible way to control the generalization ability of the boosted classifier than early stopping. Hence, in all remaining experiments, we fix the number of boosting iterations and cross-validate the regularization gain. This regularization strategy has one additional property of interest. As can be seen in the bottom

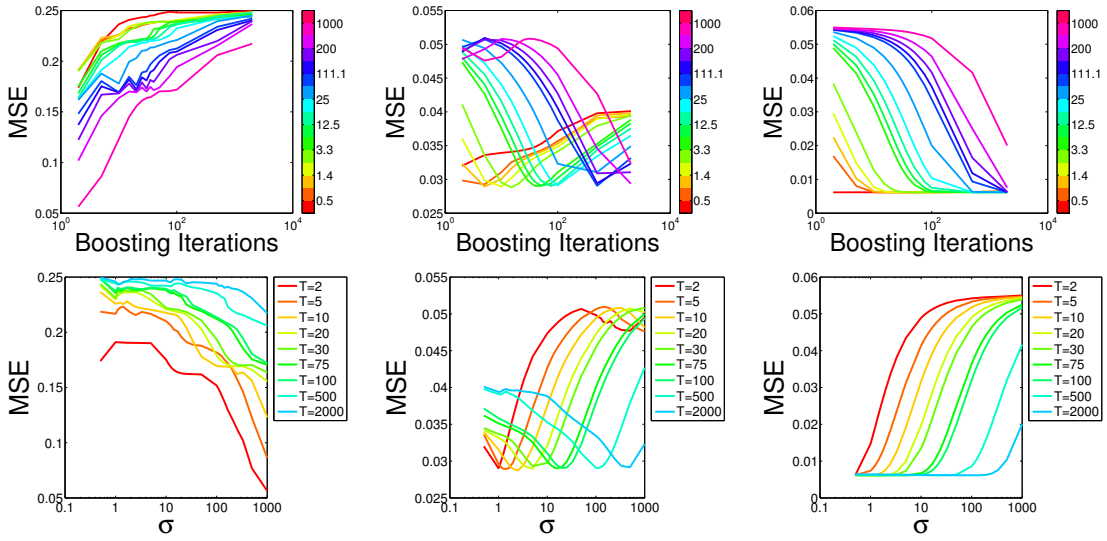


Figure 9: Top: MSE as a function of the number of boosting iterations T for different regularization gains. Bottom: MSE as a function of regularization gain σ for different numbers of boosting iterations T. From left to right: small, moderate sized, and large samples.

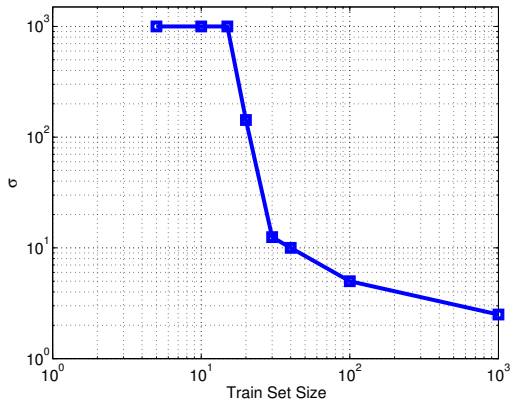


Figure 10: Cross-validated regularization gain as a function of training set size.

row of Figure 9, when the number of iterations T is fixed, the best performing regularization gain decreases with the sample size. This suggests that, when T is fixed, the cross-validated σ can be seen as a diagnostic of whether the classifier would benefit from the collection of further training data. Small samples (left of the figure) require large σ, while a small σ is sufficient for large samples (right). This effect is illustrated in Figure 10, which presents a plot of the cross-validated regularization gain as a function of training set size. Note the monotonic relation between the two variables, suggesting that regularization gain can be used as a diagnostic for data scarcity. While a large σ suggests that it is worth collecting more training data, a small σ indicates that such an effort is likely not justified. This can help learning practitioners perform cost-benefit analysis of their data collection efforts.

7.2 The Role of the Link Function

The next set of experiments used ten binary UCI data sets of relatively small size: (#1) sonar, (#2) breast cancer prognostic, (#3) breast cancer diagnostic, (#4) original Wisconsin breast cancer, (#5) Cleveland heart disease, (#6) tic-tac-toe, (#7) echo-cardiogram, (#8) Haberman's survival, (#9) Pima-diabetes, and (#10) liver disorder. These experiments aimed to evaluate the impact of of the choice of regularization (link) function on calibration and classification accuracy. Since, as discussed in Section 6.3, canonical losses implement all regularization behaviors possible for tunable regularization losses, we only considered the losses of Table 2 in these experiments. Each data set was split into five folds, four of which were used for training and one for testing. This created four train-test pairs per data set, over which the results were averaged. In all experiments, three of the four training folds were used for classifier training and one as validation set for parameter selection.

BoostLR was run for 50 iterations, using histogram-based weak learners and regularization gains $\sigma \in [0.3, 500]$. Classification accuracy was measured with test error. Since the true posterior probabilities are not known for the UCI data sets, calibration cannot be evaluated with (54). A measure of calibration commonly used when this is the case is the cross-entropy between the distributions of the true η and estimated posterior probabilities $\hat{\eta}$ (Niculescu-Mizil and Caruana, 2005). Assuming the quantization of all probabilities into K probability bins, this is defined as

$$H(\eta, \hat{\eta}) = -\sum_{k=1}^{K} p(\eta = k) \log p(\hat{\eta} = k) = -E_{\eta}[\log p(\hat{\eta})].$$

For large samples, the cross-entropy can be estimated with

$$H(\eta, \hat{\eta}) = -\sum_{i=1}^{N} \frac{1}{N} \log p(\hat{\eta}(x_i)).$$

This measure is largest for poorly calibrated classifiers that produce bimodally distributed posterior estimates, concentrated around $\hat{\eta} = 0$ and $\hat{\eta} = 1$, and smallest for well calibrated classifiers whose distribution of posteriors is less concentrated, and spread more evenly between zero and one (Niculescu-Mizil and Caruana, 2005; Mease and Wyner, 2008).

Figure 11 presents curves of the average calibration and classification ranks of the predictor designed with the GLog loss for each σ. Similar curves were obtained for all losses

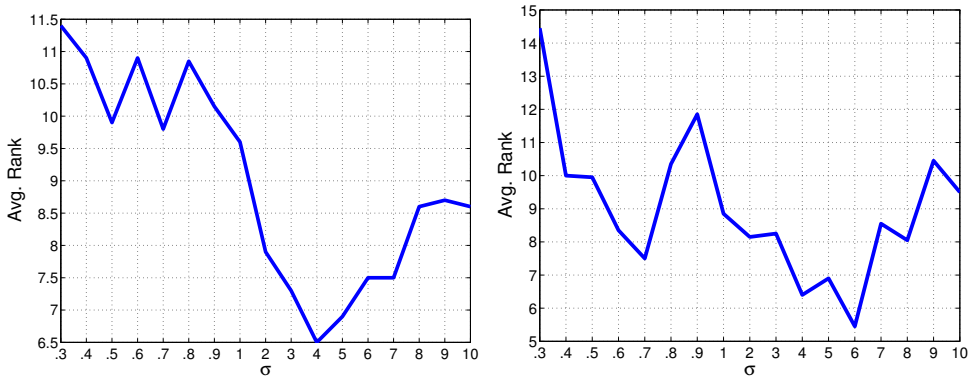


Figure 11: Average calibration (left) and classification (right) rank as a function of regularization gain for the GLog loss on the UCI data.

of Table 2. To produce these plots, a predictor was trained per data set, for 17 values of $\sigma \in [0.3, 10]$. The results were then ranked, and rank 1 (17) assigned to the value of σ of smallest (largest) cross-entropy or classification error. The ranks of each σ were then averaged over the ten data sets (Demšar, 2006). Note that the curves of classification accuracy and cross entropy rank have similar shape, although the rank curve is smoother for cross-entropy. This is because the classifier produces binary decisions by thresholding the predictor output. Nevertheless, the two plots support the conclusion that the best values of σ for these data sets are in the range of $4 \leq \sigma \leq 6$. Note that the average calibration rank for this range (between 6.5 and 7.5), is substantially better than that (more than 9.5) of the logistic loss of Figure 1 (which is identical to GLog with $\sigma = 1$). For classification, the difference is similar (between 5.5 and 6.5 for $4 \leq \sigma \leq 6$, around 9 for $\sigma = 1$). In summary, regularization strength can have a significant impact in both classification and calibration performance. The fact that best results occur for relatively large regularization gains is not surprising, given that these data sets are relatively small.

We next attempted to quantify the intrinsic regularization gain of each data set, i.e. the regularization gain that leads to best performance on that data set across all losses, and the benefits of using that regularization over the standard values (e.g. $\sigma = 1$ for the logistic loss in LogitBoost). For this, we averaged the performance of all BoostLR classifiers learned with the four losses of Table 2, for each value of σ and data set. We then determined the gain σ_{opt} of smallest average classification error per data set. This can be seen as a loss-independent measure of the intrinsic regularization gain of the data set. The associated classification error is a loss-independent estimate of the performance of a classifier tuned to this intrinsic regularization value. These results are summarized in Table 3 (top). For comparison, we also present the results of AdaBoost, LogitBoost (GLog loss with $\sigma = 1$), the average performance of BoostLR with the four losses of Table 2 when the bandwidth is constrained to $\sigma = 1$, and the drop in classification error due to the tuning to the intrinsic regularization gain of the data set. To compute this drop, we defined as ϵ_1 the average error of the BoostLR methods with the intrinsic gain, as ϵ_2 the smallest error of all other

UCI data set#	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
Classification										
AdaBoost	**11.4**	15.2	9.2	6	11.4	21.6	7.4	23.2	42.8	26.6
LogitBoost($\sigma = 1$)	12.4	15.4	8.6	5.6	11.4	46	7.2	25	40.4	**26.4**
Avg. BoostLR($\sigma = 1$)	13.25	16.4	8.06	5.53	11.6	47.95	7.15	24.6	40.65	27.4
Avg. BoostLR(σ_{opt})	11.6	**14.95**	**6.93**	**4.86**	**11.1**	**13.25**	**6.7**	**14.6**	**38.8**	26.5
Drop(%)	−1.75	1.64	14.08	12.11	2.63	38.65	6.29	37.06	3.96	−0.37
Calibration										
AdaBoost	4.70	4.40	5.31	5.58	3.89	3.453	3.77	**3.593**	3.43	3.54
LogitBoost($\sigma = 1$)	4.73	4.06	5.16	5.49	3.68	**3.414**	3.71	3.609	3.42	3.58
Avg. BoostLR($\sigma = 1$)	4.25	3.88	5.20	5.63	3.77	3.419	3.68	3.599	3.41	3.65
Avg. BoostLR(σ_{opt})	**3.71**	**3.83**	**4.48**	**4.82**	**3.58**	**3.414**	**3.50**	3.595	**3.39**	**3.53**
Drop(%)	58.2	8.8	37.3	30.8	29.2	0.0	48.2	−0.7	26.3	5.3

Table 3: Intrinsic gain of regularization, in terms of classification error (top) and probability estimation accuracy (bottom), on various UCI data sets. Avg. BoostLR(σ) is the average error of classifiers learned with the margin losses of Table 2, for regularization bandwidth σ. σ_{opt} is the bandwidth of smallest average error.

methods, and the drop as $(1 - \frac{\epsilon_1}{\epsilon_2}) \times 100\%$. Note that BoostLR($\sigma_{opt}$) outperformed all other approaches in 8 out of the 10 data sets, virtually tied the best approach in one, and performed slightly worse than the best method (AdaBoost) in another. On four of the data sets its relative drop in classification error was larger than 10% and in two larger than 30%. Note also that the averaging over the four losses does not give an unfair advantage to BoostLR (σ_{opt}), since the same average for BoostLR($\sigma = 1$) has performance equivalent to LogitBoost (which uses one of the four losses of unit gain). A similar analysis is presented in the bottom half of Table 3 for calibration performance. In this case, the drop is defined as $(1 - \frac{H_1 - H}{H_2 - H}) \times 100\%$ where H_1 is the average cross entropy of the BoostLR methods with the intrinsic gain, H_2 the smallest cross entropy of all other methods and H the entropy (minimum possible cross entropy value) of the problem. BoostLR (σ_{opt}) outperformed all other approaches in 8 out of the 10 data sets with a relative drop in cross entropy of more than 10% on six, more than 30% on four and more than 40% on two data sets. These results show that, for an equal amount of parameter regularization (all classifiers have the same number of weak learners) there can be substantial gains in tuning the regularization strength of the loss.

We next evaluated the performance of the individual regularization losses. Since they are canonical, this is equivalent to comparing the associated link $f^*_{\phi_\sigma}$ or regularization strength ρ_{ϕ_σ} functions of (45). Given that the the number of boosting iterations is the same for all methods, i.e. all classifiers have the same amount of parameter regularization, this comparison is indicative of the effectiveness of the different link functions as probability regularizers. The top half of Table 4 presents the average test error obtained for each UCI data set and loss. Also shown are the baseline results of AdaBoost and LogitBoost (GLog loss with $\sigma = 1$). The last two columns present two statistics, reporting to the number of wins of each algorithm. This is the number of data sets in which the algorithm outperformed a set

UCI data set#	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	W_1	W_2
Classification												
AdaBoost	11.4	**11.4**	9.4	6.4	14	28	6.6	21.8	41.2	28.2	-	1
LogitBoost	11.6	12.4	10	6.6	13.4	48.6	6.8	21.2	39.6	28.4	-	0
GLog	**11.2**	**11.4**	**8**	5.6	**12.4**	11.8	7	18.8	38.2	**27**	9	5
GBoost	12.6	11.6	21	18.6	17.6	**7.2**	**6**	21.8	**37.6**	28.6	3	3
GGauss	13.6	14.4	9	6	13	8.8	7.6	**18.4**	38.4	30.6	6	1
GLaplace	12	12.8	9	**5**	**12.4**	8.2	6.6	20.8	40.6	31.6	6	2
BoostLR wins	1	1	3	3	3	4	2	3	3	1	-	-
Drop (%)	1.7	0	14.9	21.9	7.5	74.3	9.1	13.2	5.0	4.2	-	-
Calibration												
AdaBoost	4.59	4.19	5.47	3.94	5.77	3.61	4.71	3.48	3.442	3.461	-	0
LogitBoost	4.75	3.85	5.47	3.861	5.65	**3.57**	4.64	3.426	3.438	3.48	-	1
GLog	4.20	3.46	4.59	3.80	5.42	3.67	3.89	3.421	**3.40**	3.49	8	1
GBoost	**3.77**	4.60	5.33	**3.69**	**5.21**	3.65	3.83	3.406	3.41	**3.44**	8	4
GGauss	4.07	**3.44**	4.70	3.71	5.49	3.62	3.87	3.429	3.439	3.53	6	1
GLaplace	3.81	3.48	**4.58**	3.76	5.31	3.63	**3.81**	3.41	3.42	3.45	9	2
BoostLR wins	4	3	4	4	4	0	4	3	3	2	-	-
Drop (%)	64.52	76.53	41.56	30.18	19.11	−6.50	62.76	22.23	28.52	13.01	-	-

Table 4: Cross validated classification error (top) and cross entropy (bottom) for each loss function and UCI data set. W_1 : number of wins over AdaBoost and LogitBoost. W_2 : number of wins over all methods.

of competitors. The two statistics differ in the composition of this set. W_1 compares the performance of each tunable regularization loss to the AdaBoost and LogitBoost baselines, evaluating how frequently each version of BoostLR outperforms the well established boosting methods. W_2 uses all other algorithms in the table as competitors, measuring how many times each algorithm achieved the best performance among all methods considered. Finally, the last two rows report similar statistics per data set. The row before last reports the number of BoostLR algorithms that outperformed both AdaBoost and LogitBoost. The last row presents the drop in test error between the established boosting methods and BoostLR. To compute this drop, we found the smallest test error ϵ_1 of Ada and LogitBoost, the smallest test error ϵ_2 of all BoostLR methods, and defined the drop as $(1 - \epsilon_2/\epsilon_1) \times 100\%$.

Several conclusions can be drawn from the table. First, statistic W_1 shows that BoostLR with either the GLog, GGauss, or GLaplace losses, beats both AdaBoost and LogitBoost in at least half of the data sets. Best performance was achieved by GLog, which beat the established methods in 9 out 10 data sets. Second, statistic W_2 shows that, while BoostLR with the GLog loss (logistic link) has the overall best performance, different links perform best for different data sets (3 overall wins for GLaplace, 2 for GBoost, and 1 for GGauss). Third, the gains of tunable loss regularization vary substantially from data set to data set. This is clear from the last two rows of the table, where BoostLR is shown to have modest improvements (less that 5% drop in error rate) for 3 data sets, significant gains (between 5 and 20% drop) in 5, and massive gains (above 20%) in 2. In general, the magnitude of the gain is correlated with the number of BoostLR variants that beat AdaBoost and

LogitBoost, e.g. the more variants beat the established methods the largest the drop in classification error. This suggests that the regularization gains of AdaBoost and LogitBoost are severely mistuned for these data sets.

The bottom half of Table 4 presents a similar analysis for calibration performance, using the cross entropy criteria. In this case the drop is defined as $(1 - \frac{H_1 - H}{H_2 - H}) \times 100\%$, where H_1 is the smallest cross entropy of Ada and LogitBoost, H_2 the smallest cross entropy of all BoostLR methods and H the entropy (minimum possible cross entropy value) of the problem. The cross entropy criteria produced similar results in terms of number of wins, but the drop in relative cross entropy was much more substantial, with a drop of more than 10% on nine data sets, more than 20% on seven, more than 40% on four and more than 60% on three.

We next evaluated the impact of the link function in the recovery of posterior probabilities. For this, we performed a comparison between BoostLR with shrinkage loss and GradientBoost + shrinkage. As discussed in Section 6.4, while the two algorithms produce identical classifiers, the posterior probability estimates are not the same. GradientBoost relies on (12), BoostLR uses (51). The probabilities recovered, using the GLog loss, on the ten UCI data sets were compared. In the first set of experiments, the regularization gain of BoostLR was fixed at $\sigma = 10$ and the learning rate of shrinkage at $\lambda = 0.1$. The calibration performance of both algorithms is shown, for each data set, in the top half of Table 5. BoostLR has considerably better calibration on all ten data sets. We also compared the results achieved with cross-validation of the regularization gain of BoostLR and the learning rate of shrinkage. As shown on the bottom half of Table 5, BoostLR has better calibration on seven of the ten data sets. In summary, even for shrinkage losses, where BoostLR and GradientBoost with shrinkage produce identical classifiers, the fact that BoostLR uses the correct link for probability recovery enables it to achieve superior calibration performance.

7.3 The Role of the Binding Function

The following set of experiments aimed to evaluate the impact of the binding function. For this, we considered the scenario where BoostLR differs from GradientBoost with shrinkage even for classification, by using the α-tunable loss of (53). As discussed in Section 6.5, the additional α parameter of this loss enables independent control of binding and link functions. This allows the loss to adapt to the outlier content of the data. To evaluate the benefits of this adaptation, we compared the classification and calibration performance of BoostLR with the loss of (53) to that of AdaBoost with shrinkage. All experiments relied on five-fold cross-validation. For both algorithms the regularization gain σ was cross-validated among 10 values in $[1, 10]$. The α parameter of BoostLR was cross-validated among 5 values in $[0, 1/2]$. Various percentages of outliers were added to the ten UCI data sets by randomly flipping labels of training examples. The classification and calibration performance of the two algorithms are presented in Figure 12. The figure depicts the average rank of the classifiers learned by the two methods, over the ten UCI data sets, as a function of the percentage of outliers. BoostLR has better calibration (smaller rank) for all outlier percentages. This illustrates the benefits of α-tuning for noisy data. For classification, the same holds for all outlier percentages other than 15%. The reversal of ranks for this percentage can be explained by the noisier nature of the classification data

UCI data set#	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
Fixed $\sigma = 10$ ($\lambda = 0.1$)										
BoostLR	**4.13**	**3.91**	**4.56**	**5.37**	**3.47**	**3.58**	**3.73**	**3.84**	**3.65**	**4.13**
Shrinkage	4.65	4.49	5.30	5.74	4.63	4.97	4.16	4.35	4.97	4.19
Cross validated σ and λ										
BoostLR	**4.19**	**3.89**	**4.59**	**5.38**	**3.46**	3.45	**3.80**	3.62	**3.40**	3.53
Shrinkage	4.66	4.52	5.30	5.70	3.85	**3.42**	3.87	**3.59**	3.43	**3.46**

Table 5: Calibration performance (cross-entropy) of BoostLR and GradientBoost with shrinkage on the UCI data.

(due to the hard decision made by the classifier). Even though the BoostLR classifiers are better calibrated, the classification error is larger. We note that better results should be possible with α-tunable losses that implement binding functions expressly designed to achieve outlier robustness, e.g. that of the Savage loss (Masnadi-Shirazi and Vasconcelos, 2008). This is left for future work. The goal here was not to produce the classifier of greatest possible robustness, only to investigate the benefits of independently controlling the link and binding functions.

7.4 Experiments on Larger Data sets

The data sets used in the previous section are of relatively small size. To investigate the benefits of loss regularization for larger data sets, we considered the ADULT, LETTER.p1 and LETTER.p2 data sets, which are widely used for comparing ensemble methods (Niculescu-Mizil and Caruana, 2005; Caruana et al., 2004). Missing values in the ADULT training and testing sets were omitted, leading to 30,162 training examples, of which 7,508 are positive and 22,654 negative. The test set consists of 15,060 examples, of which 3,700 are positive and 11,360 negative. The LETTER data was converted into two binary data sets (Caruana et al., 2004). The LETTTER.p1 data set treats the confusable letter "O" as the positive class, and the remaining 25 letters of the alphabet as the negative class, resulting in a highly unbalanced classification problem. LETTER.p2 uses the first 13 letters of the alphabet as the negative class and the last 13 as the positive class, resulting in a balanced but difficult problem. Both datasets contain 4,000 training and 16,000 test examples. As before, all classifiers were learned with BoostLR, using histogram weak learners, and cross-validation of the regularization gain. The performance of the GLog and GLaplacian losses was compared to that of the exponential loss, used by AdaBoost, and GLog with unit gain, used by LogitBoost. Each boosting algorithm was run for 100 iterations.

Table 6 presents the error achieved by each method, and the corresponding regularization gain. Note that 1) best performance was never attained with the logistic loss (GLog with $\sigma = 1$) of LogitBoost, or the exponential loss of AdaBoost, 2) each of the two losses of tuned gain outperformed both standard boosting losses, and 3) in each case the gains were substantial. Note also that the optimal σ was always smaller than one. This is explained by the larger size of the datasets used in this experiment. The optimality of small σ in this experiment and larger σ in the experiments of the previous section is in agreement with

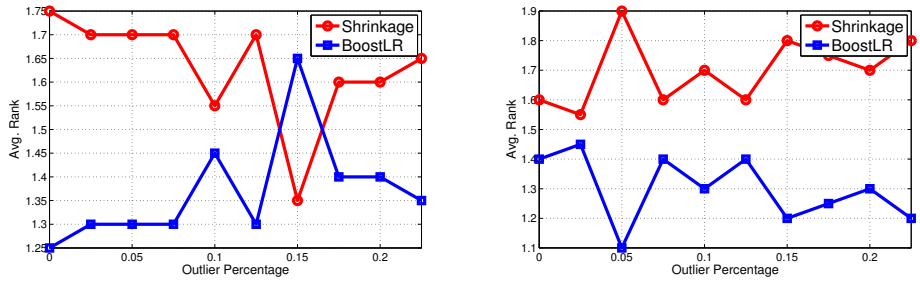


Figure 12: Average classification (left) and calibration (right) rank as a function of percentage of outliers on the UCI data, for BoostLR and AdaBoost with shrinkage.

the observations of Section 7.1. To further investigate this point, we considered reduced versions of LETTER.p2, by randomly subsampling training examples. More precisely, the training set was subsampled by a factor of 2 (DIV2) and 4 (DIV4). The size of the test set was not changed. Table 7 presents 1) the optimal regularization gain for each loss, and 2) the difference between the number of testing errors produced by the exponential and each of the regularization losses, for each training set size. Note how 1) the regularization gain increases for smaller datasets, eventually becoming larger than one, and 2) the classification gains are larger for the smaller datasets. As previously noted in Section 7.1, these results suggest that large margins are important for small datasets but do not add much, to classifier performance, for large ones.

8. Conclusion

Large margins and parameter regularization are commonly used to assure classifier generalization. Large margins are implemented with risks based on margin losses, regularization by inclusion, in these risks, of terms that encourage parameter sparsity. In this work, we have shown that margin losses can also be viewed as regularizers of posterior class probability estimates. In fact, an analysis of both 1) probability estimation error, and 2) generalization bounds, has shown that, for proper losses of generalized logit link, loss-based regularization amplifies the strength of parameter regularization by a factor equal to the loss margin. These losses were also shown to have a simple decomposition in terms of a link and a binding function. The link determines the loss behavior around the classification boundary and is responsible for its regularization strength. The binding function determines the loss behavior for large margins and is responsible for its outlier robustness. In this way, link and binding functions partition the space of losses into equivalence classes of identical probability regularization or outlier robustness. These equivalence classes are isomorphic to the set of symmetric scale probability densities of unique maximum at the origin and the set of monotonically decreasing odd functions, respectively. Each equivalence class contains many tunable regularization losses, parameterized by a regularization gain σ.

Tunable regularization losses can be used to derive boosting algorithms with loss regularization (BoostLR) of tunable strength. Three classes of losses were considered in this

UCI data set	ADULT		LETTER1		LETTER2	
	error	σ	error	σ	error	σ
GLog	**2406**	0.25	427	0.33	**2831**	0.5
GLaplacian	2680	0.45	**420**	0.25	2844	0.3
Exponential	2696		529		2940	
Logit ($\sigma = 1$)	2673		464		2867	

Table 6: Optimal regularization gain and corresponding classification error on the large UCI datasets.

LETTER2	DIV1	DIV2	DIV4
GLog	109	179	260
	$\sigma = 0.5$	$\sigma = 1.66$	$\sigma = 2$
GLaplacian	96	178	186
	$\sigma = 0.3$	$\sigma = 1$	$\sigma = 2$

Table 7: Optimal σ as a function of training set size and corresponding classification error gain over exponential loss.

work: 1) canonical losses, which have linear binding functions and no flexibility in terms of outlier modeling, 2) shrinkage losses, which support equally parameterized link and binding function pairs, and 3) α-tunable losses, which enable independent parameterization of link and binding function. BoostLR algorithms with shrinkage losses were then shown to implement the well known shrinkage procedure. This offers an alternative explanation of shrinkage as regularization of posterior probability estimates, explaining its success in terms of large margins and generalization bounds. On the other hand, the flexibility of α-tunable losses enabled the derivation of a boosting algorithm that generalizes both AdaBoost and LogitBoost, behaving as either of them according to the data to classify.

Extensive experiments on a series of synthetic and UCI datasets showed that, when the regularization gain is optimized, BoostLR can substantially outperform previous boosting algorithms, with respect to both classification error and probability calibration. These results challenge the popular belief that large-margin classifiers are not capable of producing calibrated probability estimates. They also shed some light on the synergies between loss-based and parameter regularization in boosting algorithms, where parameter regularization is usually implemented by early stopping. For small samples, which demand strong regularization, this can be insufficient, and a large loss regularization gain required. For large samples, where little regularization is necessary, the bias introduced by the combination of parameter and loss regularization can be too large. Better results can be obtained by weakening the regularization. This can be accomplished by using a smaller σ.

Appendix A. Relations Between Loss Margin and Regularization Strength

In this appendix, we determine the conditions under which the loss margin μ_ϕ of (25) is a measure of the regularization strength of the loss ϕ.

Lemma 9 *Let $\phi(v)$ be a twice differentiable proper loss of monotonically increasing inverse link $[f^*_\phi]^{-1}(\eta)$. Then (26) holds. Furthermore, $[f^*_\phi]^{-1}(\eta)$ has an inflection point at the origin. If this inflection point is the maximum of $\{[f^*_\phi]^{-1}\}'(v)$, then the regularization strength is lower bounded by twice the loss margin, as in (27), and $\phi(v)$ is a regularization loss if and only if $\mu_\phi \geq \frac{1}{2}$.*

Proof If ϕ is proper, it follows from (24) that

$$\begin{aligned}
\phi'(v) &= \left(1 - [f_\phi^*]^{-1}(v)\right) [C_\phi^*]'' \left([f_\phi^*]^{-1}(v)\right) \{[f_\phi^*]^{-1}\}'(v) \\
\phi''(v) &= -\left(\{[f_\phi^*]^{-1}\}'(v)\right)^2 [C_\phi^*]'' \left([f_\phi^*]^{-1}(v)\right) \\
&+ \left(1 - [f_\phi^*]^{-1}(v)\right) [C_\phi^*]^{(3)} \left([f_\phi^*]^{-1}(v)\right) \left(\{[f_\phi^*]^{-1}\}'(v)\right)^2 \\
&+ \left(1 - [f_\phi^*]^{-1}(v)\right) [C_\phi^*]'' \left([f_\phi^*]^{-1}(v)\right) \{[f_\phi^*]^{-1}\}''(v).
\end{aligned}$$

From (22) and (23), $[f_\phi^*]^{-1}(0) = 1/2$, $[C_\phi^*]^{(3)}(\eta) = -[C_\phi^*]^{(3)}(1-\eta)$, and $\{[f_\phi^*]^{-1}\}''(v) = -\{[f_\phi^*]^{-1}\}''(-v)$, and it follows that

$$\{[f_\phi^*]^{-1}\}''(0) = 0 \tag{55}$$

$$[C_\phi^*]^{(3)}\{[f_\phi^*]^{-1}(0)\} = 0, \tag{56}$$

from which $\phi'(0) = \frac{1}{2}[C_\phi^*]'' \left(\frac{1}{2}\right) \{[f_\phi^*]^{-1}\}'(0)$, $\phi''(0) = -\left(\{[f_\phi^*]^{-1}\}'(0)\right)^2 [C_\phi^*]'' \left(\frac{1}{2}\right)$, and

$$\mu_\phi = \frac{\{[f_\phi^*]^{-1}\}'(0)}{2\left(\{[f_\phi^*]^{-1}\}'(0)\right)^2} = \frac{\rho_\phi(0)}{2}.$$

Furthermore, from (55), $[f_\phi^*]^{-1}$ has an inflection point at the origin. From (14), if this point is a maximum of $\{[f_\phi^*]^{-1}\}'$, then $\rho_\phi(v) \geq \rho_\phi(0)$ for all v, (27) holds, and the theorem follows. ■

Appendix B. The Generalized Logit Link

In this appendix, we discuss some properties of the generalized logit link that are used in the remaining results of this work.

B.1 Properties

We start by noting that the conditions of Definition 2 are a set of sufficient conditions for a function to be the link of a proper loss. The monotonicity of Property 1. is sufficient for the invertibility of π. While it is not necessary that π^{-1} be increasing, this guarantees that the probability estimates $\eta = \pi^{-1}(p)$ increase with p. Property 2. and 3. suffice for π to be a link of some proper loss. Property 3. is the condition of (23). When combined with 1. and 2. it constrains $\pi^{-1}(v)$ to be in $[0, 1]$. This guarantees that η is a probability. While Property 3. is necessary, this is not the case of Property 2. For example,

$$\pi^{-1}(v) = \frac{1+v}{2}, \qquad v \in [-1, 1]$$

is a valid inverse link. However, the use of such a link requires that $p(\mathbf{x}) \in [-1, 1]$ for $\eta(\mathbf{x}) = \pi^{-1}(p(\mathbf{x}))$ to be a probability. This constraint on $p(\mathbf{x})$ has to be enforced by learning algorithms, complicating the underlying optimization. We are aware of no benefit

in adopting such a link over a generalized logit. Property 2. eliminates all links of this type. Finally, Property 4. is necessary and sufficient for π^{-1} to have a unique inflection point at the origin. Note that the if statement follows from Property 3. but not the only if. A "staircase" of sigmoids could satisfy 1.-3. and have multiple inflection points. Property 7. of the following lemma shows that this suffices for the inverse of the generalized logit to have maximum derivative at the origin. It follows that all conditions of Lemma 9 hold when $f_\phi^*(\eta)$ is a generalized logit link, proving Theorem 4.

Lemma 10 *A generalized logit π has the following properties*

1. $\pi^{-1}(v) \in (0,1)$
2. $\lim_{v\to-\infty} \pi^{-1}(v) = 0$
3. $\pi^{-1}(0) = .5$
4. $(\pi^{-1})^{(n)}(-v) = (-1)^{n+1}(\pi^{-1})^{(n)}(v)$
5. $(\pi^{-1})^{(n)}(0) = 0$, *whenever n is even*
6. $\lim_{v\to\pm\infty}(\pi^{-1})^{(n)}(v) = 0, n \geq 1.$
7. $(\pi^{-1})'(v)$ *has a unique maximum at the origin.*

Proof Properties 1.-5. are a straightforward consequence of Properties 1.-3. of Definition 2. Property 6. follows from the fact that π^{-1} is monotonically increasing and lower and upper bounded by 0 and 1, respectively. Property 7. then follows from the fact that $(\pi^{-1})'$ is positive for all v and only has one critical point at the origin, by Property 4. of Definition 2. ■

B.2 Parametric Generalized Logit Links

In this section we show that the set $\mathcal{L}$ of generalized logit links is isomorphic to a set of probability density functions.

Lemma 11 *The set $\mathcal{L}$ of parametric generalized logit links of (38) is isomorphic to the set of parametric continuous scale probability density functions (pdfs)*

$$\psi_\sigma(v) = \frac{1}{\sigma}\psi\left(\frac{v}{\sigma}\right),$$

where $\psi(v)$ has unit scale, a unique maximum at the origin, and $\psi(-v) = \psi(v)$.

Proof Let $c(v) = \int \psi(v)dv$ be the cdf of a continuous scale pdf $\psi(v)$. Then $c(v)$ satisfies Properties 1. and 2. of Definition 2. Property 3. is also met if $\psi(v)$ has symmetry $\psi(-v) = \psi(v)$, and Property 4. if $\psi(v)$ has a unique maximum at the origin. Finally, from the continuity of $\psi(v)$, $c(v)$ has an inverse and $c^{-1}(v)$ is a generalized logit link. Since any generalized logit link with the properties of Definition 2 defines one such cdf, the set of

generalized logit links is isomorphic to the set of continuous scale pdfs $\psi(v)$ of symmetry $\psi(-v) = \psi(v)$ and a unique maximum at the origin.

Let $\psi(v)$ be the pdf corresponding to $f^*_\phi(\eta)$, i.e. $[f^*_\phi]^{-1}(v) = \int_{-\infty}^{v} \psi(q)dq$. Then, for any σ, it follows from (38) that

$$[f^*_{\phi_\sigma}]^{-1}(v) = [f^*_\phi]^{-1}\left(\frac{v}{\sigma}\right)$$

is the cdf of $\psi_\sigma(v)$, as defined in (44). Since this procedure can be repeated for any link function $f^*_\phi(\eta)$, $\mathcal{L}$ is isometric to the set of these pdfs. ■

Appendix C. The Binding Function

In this appendix, we discuss the properties of the binding function.

Lemma 12 *Let $\beta_\phi(v)$ be the binding function of a proper loss $\phi(v)$ of generalized logit link $f^*_\phi(\eta)$, and minimum risk $C^*_\phi(\eta)$. Then*

1. *the behavior of $\phi(v)$ for $v \to \pm\infty$ is determined by $\beta_\phi(v)$.*
2. *$\beta_\phi(v)$ is monotonically decreasing.*
3. *the mapping $[C^*_\phi]'(\eta) = \beta_\phi\left(f^*_\phi(\eta)\right)$ is one-to-one.*
4. *$\beta_\phi(v)$ is an odd function, i.e. $\beta_\phi(-v) = -\beta_\phi(v)$.*

Proof To prove Property 1. we note that, combining (31) with Properties 2. of Definition 2 and Lemma 10, and $C^*_\phi(0) = C^*_\phi(1) = 0$, it follows that

$$\lim_{v\to\pm\infty} \phi(v) = \lim_{v\to\pm\infty} (1 - [f^*_\phi]^{-1}(v))[C^*_\phi]'\{[f^*_\phi]^{-1}(v)\}.$$

The property follows from the fact that $\lim_{v\to\pm\infty}(1-[f^*_\phi]^{-1}(v)) \in \{0, 1\}$ and (34). Property 2 follows from the fact that

$$\beta'_\phi(v) = [C^*_\phi]''\left([f^*_\phi]^{-1}(v)\right)\{[f^*_\phi]^{-1}\}'(v)$$

C^*_ϕ is concave (Theorem 3) and $\{[f^*_\phi]^{-1}\}'(v) > 0$ (Property 1 of Definition 2). Property 3 then follows from (34) and Property 2. Finally, Property 4 follows from

$$\begin{aligned}
\beta_\phi(-v) &= [C^*_\phi]'\left([f^*_\phi]^{-1}(-v)\right) \\
&= [C^*_\phi]'\left(1 - [f^*_\phi]^{-1}(v)\right) \\
&= -[C^*_\phi]'\left([f^*_\phi]^{-1}(v)\right) = -\beta_\phi(v).
\end{aligned}$$

where we have used (22) and (23). ■

Appendix D. Properties of Proper Losses

In this appendix, we derive various properties of proper losses.

D.1 Proper Losses of Generalized Logit Link

The following lemma summarizes various properties of proper losses with generalized logit link.

Lemma 13 *Let $\phi(v)$ be a proper loss of generalized logit link $f^*_\phi(\eta)$ and binding function $\beta_\phi(v)$. Then, the following properties hold.*

1. *$\phi(v)$ is monotonically decreasing*
2. *$\phi(v)$ is convex if and only if*

$$\frac{\beta''_\phi(v)}{\beta'_\phi(v)} < \frac{\{[f^*_\phi]^{-1}\}'(v)}{(1-[f^*_\phi]^{-1}(v))}, \quad \forall v \tag{57}$$

3. $\lim_{v\to-\infty}\phi(v) = \lim_{v\to-\infty}\beta_\phi(v)$
4. $\phi'(0) = \frac{1}{2}\beta'_\phi(0)$
5. $\phi''(0) = -\frac{\beta'_\phi(0)}{\rho_\phi(0)}$.

Proof Property 1. follows from (35) and the facts that $(1-[f^*_\phi]^{-1}(v)) > 0$ (Properties 1. and 2. of Definition 2) and $\beta'_\phi(v) < 0$ (Property 2. of Lemma 12). To prove Property 2. we take derivatives on both sides of (35),

$$\phi''(v) \;=\; -\{[f^*_\phi]^{-1}\}'(v)\beta'_\phi(v) + (1-[f^*_\phi]^{-1}(v))\beta''_\phi(v).$$

It follows that $\phi(v)$ is convex if and only if, for all v, $\{[f^*_\phi]^{-1}\}'(v)\beta'_\phi(v) < (1-[f^*_\phi]^{-1}(v))\beta''_\phi(v)$. Since $(1-[f^*_\phi]^{-1}(v)) > 0$ and $\beta'_\phi(v) < 0$, this is identical to (57). Property 3. follows from (36) and Property 2. of Lemma 10, since $\lim_{v\to-\infty}\phi(v) = C^*_\phi(0) + \lim_{v\to-\infty}(1-[f^*_\phi]^{-1}(v))\beta_\phi(v)$, $C^*_\phi(0) = 0$, and $\lim_{v\to\infty}(1-[f^*_\phi]^{-1}(v)) = 1$. Property 4. is a simple consequence of (23), which implies that $[f^*_\phi]^{-1}(0) = \frac{1}{2}$. Finally, Property 5. follows from $\phi''(0) = -\{[f^*_\phi]^{-1}\}'(0)\beta'_\phi(0) + \frac{1}{2}\beta''_\phi(0)$ and Property 4. of Lemma 12, which implies that $\beta''_\phi(0) = 0$. ∎

D.2 Canonical Regularization Losses

The following lemma summarizes various properties of canonical regularization losses.

Lemma 14 *Let $\phi_\sigma(v)$ be a tunable regularization loss of binding function as in (46). The following properties hold.*

1. $\phi''_\sigma(v) > 0, \forall v$
2. $\lim_{v\to\infty} \phi'_\sigma(v) = 0$
3. $\lim_{v\to-\infty} \phi'_\sigma(v) = -1$
4. $\phi'_\sigma(0) = -1/2$
5. *ϕ''_σ is maximum at the origin.*
6. *the loss margin and regularization strength are related by* $2\mu_{\phi_\sigma} = \rho_{\phi_\sigma}(0) = \frac{1}{\phi''_\sigma(0)}$.

Proof Properties 1. and 2. follow from (47) and Properties 1. and 2. of Definition 2. Properties 3. to 5. follow from Properties 2., 3., and 7. of Lemma 10. Property 6. follows from $\mu_{\phi_\sigma} = \sigma\mu_\phi$ and the combination of (29), Property 5. of Lemma 13, and (46). ■

References

S. Avidan. Ensemble tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29:261–271, 2007.

P. Bartlett, M. Jordan, and J.D. McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101:138–156, 2006.

P.J. Bickel, Y. Ritov, and A. Zakai. Some theory for generalized boosting algorithms. *Journal of Machine Learning Research*, 7:705–732, 2006.

G. Blanchard, G. Lugosi, and N. Vayatis. On the rate of convergence of regularized boosting classifiers. *Journal of Machine Learning Research*, 4:861–894, 2003.

P. Buhlmann and T. Hothorn. Boosting algorithms: Regularization, prediction and model fitting. *Statistical Science*, 22:477–505, 2007.

P. Buhlmann and B. Yu. Boosting with the L2 Loss: Regression and Classification. *Journal of the American Statistical Association*, 98:324–339, 2003.

A. Buja, W. Stuetzle, and Y. Shen. Loss functions for binary class probability estimation and classification: Structure and applications. 2006.

R. Caruana, A. Niculescu-mizil, G. Crew, and A. Ksikes. Ensemble selection from libraries of models. In *International Conference on Machine Learning*, pages 137–144, 2004.

O. Chapelle. Training a support vector machine in the primal. *Neural Computation*, 19: 1155–1178, 2007.

K. Chen and S. Wang. Regularized boost for semi-supervised learning. In *Advances in Neural Information Processing Systems*, volume 20, pages 281–288. MIT Press, 2008.

K. Chen and S. Wang. Semi-supervised learning via regularized boosting working on multiple semi-supervised assumptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33:129–143, 2011.

M. Culp, K. Johnson, and G. Michailidis. On adaptive regularization methods in boosting. *Journal of Computational Graphics and Statistics*, 20:804–937, 2011.

M.H. DeGroot and S.E. Fienberg. The comparison and evaluation of forecasters. *The Statistician*, 32:14–22, 1983.

J. Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.

T.G. Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine Learning*, 40:139–157, 2000.

J. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29(5):1189–1232, 2001.

J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: A statistical view of boosting. *Annals of Statistics*, 28:337–407, 2000.

T. Gneiting and A.E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102:359–378, 2007.

M. Gonen, A. G. Tanugur, and E. Alpaydm. Multiclass posterior probability support vector machines. *IEEE Transactions on Neural Networks*, 19(1):130–139, 2008.

Hastie, Tibshirani, and Friedman. *The Elements of Statistical Learning*. Springer-Verlag Inc, New York, 2001.

D. Hosmer and S. Lemeshow. *Applied Logistic Regression (2nd ed.)*. John Wiley Sons Inc, New York, 2000.

X. Huang, L. Shi, and J. Suykens. Support vector machine classifier with pinball loss. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(5):984–997, 2014.

W. Jiang. Process consistency for adaboost. *Annals of Statistics*, 32:13–29, 2004.

R. Jin, Y. Liu, L. Si, J. Carbonell, and A.G. Hauptmann. A new boosting algorithm using input-dependent regularizer. In *Proceedings of Twentieth International Conference on Machine Learning*, 2003.

C. Leistner, A. Saffari, P.M. Roth, and H. Bischof. On robustness of on-line boosting - a competitive study. In *IEEE International Conference on Computer Vision and Pattern Recognition Workshop on On-line Computer Vision*, 2009.

M. Liu and B.C. Vemuri. Robust and Efficient Regularized Boosting Using Total Bregman Divergence. In *IEEE Proceedings of the 24th Conference on Computer Vision and Pattern Recognition*, pages 2897–2902, 2011.

A.C. Lozano, S.R. Kulkarni, and R.E. Schapire. Convergence and consistency of regularized boosting algorithms with stationary β-mixing observations. In *Advances in Neural Information Processing Systems*, volume 18, pages 819–826. MIT Press, 2006.

A.C. Lozano, S.R. Kulkarni, and R.E. Schapire. Convergence and consistency of regularized boosting with weakly dependent observations. *IEEE Transactions on Information Theory*, 60(1):651–660, 2014.

G. Lugosi and N. Vayatis. On the bayes-risk consistency of regularized boosting methods. *Annals of Statistics*, 32:30–55, 2004.

R. Maclin and D. Opitz. An empirical evaluation of bagging and boosting. In *In Proceedings of the Fourteenth National Conference on Artificial Intelligence*, pages 546–551. AAAI Press, 1997.

H. Masnadi-Shirazi and N. Vasconcelos. On the design of loss functions for classification: theory, robustness to outliers, and savageboost. In *Advances in Neural Information Processing Systems*, pages 1049–1056. MIT Press, 2008.

H. Masnadi-Shirazi and N. Vasconcelos. Cost-sensitive boosting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33:294–309, 2011.

L. Mason, J. Baxter, P. Bartlett, and M. Frean. Boosting Algorithms as Gradient Descent. In *Advances in Neural Information Processing Systems*, pages 512–518. MIT Press, 2000.

R. McDonald, D. Hand, and I. Eckley. An empirical comparison of three boosting algorithms on real data sets with artificial class noise. In *International Workshop on Multiple Classifier Systems*, 2003.

D. Mease and A.J. Wyner. Evidence contrary to the statistical view of boosting. *Journal of Machine Learning Research*, 9:131–156, 2008.

J.M. Moguerza and A. Munoz. Support vector machines with applications. *Statistical Science*, 21:322–336, 2006.

A. Natekin and A. Knoll. Gradient boosting machines, a tutorial. *Frontiers in Neurorobotics*, 7(21), 2013.

A. Niculescu-Mizil and R. Caruana. Obtaining calibrated probabilities from boosting. In *Uncertainty in Artificial Intelligence*, pages 413–419, 2005.

J. Platt. Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. In *Advances in Large Margin Classifiers*, pages 61–74, 2000.

G. Raskutti, M.J. Wainwright, and B. Yu. Early stopping and non-parametric regression: An optimal data-dependent stopping rule. *Journal of Machine Learning Research*, 15: 335–366, 2014.

B. Rasolzadeh, L. Petersson, and N. Pettersson. Response binning: Improved weak classifiers for boosting. In *IEEE Intelligent Vehicle Symposium*, 2006.

M. Reid and R. Williamson. Composite binary losses. *Journal of Machine Learning Research*, 11:2387–2422, 2010.

R. Rifkin and A. Klautau. In defense of one-vs-all classification. *Journal of Machine Learning Research*, 5:101–141, 2004.

S. Rosset, J. Zhu, T. Hastie, and R. Schapire. Boosting as a regularized path to a maximum margin classifier. *Journal of Machine Learning Research*, 5:941–973, 2004.

B. Saha, G. Kunapuli, N. Ray, J. Maldjian, and S. Natarajan. Ar-boost: Reducing overfitting by a robust data-driven regularization strategy. In *European Conference on Machine Learning*, pages 1–16, 2013.

L.J. Savage. The elicitation of personal probabilities and expectations. *Journal of the American Statistical Association*, 66:783–801, 1971.

R. E. Schapire, Y. Freund, P. Bartlett, and W. S. Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. *Annals of Statistics*, 26:1651–1686, 1998.

R.E. Schapire and Y. Singer. Boostexter: A boosting-based system for text categorization. *Machine Learning*, 39:135–168, 2000.

Y. Shiraishi and K. Fukumizu. Statistical approaches to combining binary classifiers for multi-class classification. *Neurocomputing*, 74(5):680–688, 2011.

V.N. Vapnik. *Statistical Learning Theory*. John Wiley Sons Inc, 1998.

P. Viola and M. Jones. Robust real-time face detection. *International Journal Computer Vision*, 57:137–154, 2004.

P. Viola, M. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. In *Ninth IEEE International Conference on Computer Vision*, volume 2, pages 734–741, 2003.

B. Wu and R. Nevatia. Simultaneous object detection and segmentation by boosting local shape feature based classifier. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 18–23, 2007.

B. Wu, H. Ai, C. Huang, and S. Lao. Fast rotation invariant multi-view face detection based on real adaboost. In *Sixth IEEE International Conference on Automatic Face and Gesture Recognition*, pages 79–84, 2004.

Y. Xi, Z. Xiang, P. Ramadge, and R. Schapire. Speed and sparsity of regularized boosting. *Journal of Machine Learning Research*, 5:615–622, 2009.

Z. Xiang, Y. Xi, U. Hasson, and P. Ramadge. Boosting with spatial regularization. In *Advances in Neural Information Processing Systems*, pages 2107–2115. MIT Press, 2009.

B. Zadrozny. Reducing multiclass to binary by coupling probability estimates. In *In Proc. Neural Information Processing Systems*, 2001.

T. Zhang. Statistical behavior and consistency of classification methods based on convex risk minimization. *Annals of Statistics*, 32:56–85, 2004.

T. Zhang and B. Yu. Boosting with early stopping: Convergence and consistency. *Annals of Statistics*, 33:1538–1579, 2005.

J. Zhu and T. Hastie. Kernel logistic regression and the import vector machine. In *Journal of Computational and Graphical Statistics*, pages 1081–1088. MIT Press, 2001.

Journal of Machine Learning Research 16 (2015) 2797-2835 Submitted 3/14; Revised 9/14; Published 12/15

Online Tensor Methods for Learning Latent Variable Models

Furong Huang FURONGH@UCI.EDU
U. N. Niranjan UN.NIRANJAN@UCI.EDU
Mohammad Umar Hakeem MHAKEEM@UCI.EDU
Animashree Anandkumar A.ANANDKUMAR@UCI.EDU
Electrical Engineering and Computer Science Dept.
University of California, Irvine
Irvine, USA 92697, USA

Editor: David Blei

Abstract

We introduce an online tensor decomposition based approach for two latent variable modeling problems namely, (1) community detection, in which we learn the latent communities that the social actors in social networks belong to, and (2) topic modeling, in which we infer hidden topics of text articles. We consider decomposition of moment tensors using stochastic gradient descent. We conduct optimization of multilinear operations in SGD and avoid directly forming the tensors, to save computational and storage costs. We present optimized algorithm in two platforms. Our GPU-based implementation exploits the parallelism of SIMD architectures to allow for maximum speed-up by a careful optimization of storage and data transfer, whereas our CPU-based implementation uses efficient sparse matrix computations and is suitable for large sparse data sets. For the community detection problem, we demonstrate accuracy and computational efficiency on Facebook, Yelp and DBLP data sets, and for the topic modeling problem, we also demonstrate good performance on the New York Times data set. We compare our results to the state-of-the-art algorithms such as the variational method, and report a gain of accuracy and a gain of several orders of magnitude in the execution time.

Keywords: mixed membership stochastic blockmodel, topic modeling, tensor method, stochastic gradient descent, parallel implementation, large datasets

1. Introduction

The spectral or moment-based approach involves decomposition of certain empirical moment tensors, estimated from observed data to obtain the parameters of the proposed probabilistic model. Unsupervised learning for a wide range of latent variable models can be carried out efficiently via tensor-based techniques with low sample and computational complexities (Anandkumar et al., 2012). In contrast, usual methods employed in practice such as expectation maximization (EM) and variational Bayes do not have such consistency guarantees. While the previous works (Anandkumar et al., 2013b) focused on theoretical guarantees, in this paper, we focus on the implementation of the tensor methods, study its performance on several datasets.

1.1 Summary of Contributions

We consider two problems: (1) community detection (wherein we compute the decomposition of a tensor which relates to the count of 3-stars in a graph) and (2) topic modeling (wherein we consider the tensor related to co-occurrence of triplets of words in documents); decomposition of the these tensors allows us to learn the hidden communities and topics from observed data.

Community detection: We recover hidden communities in several real datasets with high accuracy. When ground-truth communities are available, we propose a new error score based on the hypothesis testing methodology involving p-values and false discovery rates (Strimmer, 2008) to validate our results. The use of p-values eliminates the need to carefully tune the number of communities output by our algorithm, and hence, we obtain a flexible trade-off between the fraction of communities recovered and their estimation accuracy. We find that our method has very good accuracy on a range of network datasets: Facebook, Yelp and DBLP. We summarize the datasets used in this paper in Table 6. To get an idea of our running times, let us consider the larger DBLP collaborative data set for a moment. It consists of 16 million edges, one million nodes and 250 communities. We obtain an error of 10% and the method runs in about two minutes, excluding the 80 minutes taken to read the edge data from files stored on the hard disk and converting it to sparse matrix format.

Compared to the state-of-the-art method for learning MMSB models using the stochastic variational inference algorithm of (Gopalan et al., 2012), we obtain several orders of magnitude speed-up in the running time on multiple real datasets. This is because our method consists of efficient matrix operations which are *embarrassingly parallel*. Matrix operations are carried out in the sparse format which is efficient especially for social network settings involving large sparse graphs. Moreover, our code is flexible to run on a range of graphs such as directed, undirected and bipartite graphs, while the code of (Gopalan et al., 2012) is designed for homophilic networks, and cannot handle bipartite graphs in its present format. Note that bipartite networks occur in the recommendation setting such as the Yelp data set. Additionally, the variational implementation in (Gopalan et al., 2012) assumes a homogeneous connectivity model, where any pair of communities connect with the same probability and the probability of intra-community connectivity is also fixed. Our framework does not suffer from this restriction. We also provide arguments to show that the Normalized Mutual Information (NMI) and other scores, previously used for evaluating the recovery of overlapping community, can underestimate the errors.

Topic modeling: We also employ the tensor method for topic-modeling, and there are many similarities between the topic and community settings. For instance, each document has multiple topics, while in the network setting, each node has membership in multiple communities. The words in a document are generated based on the latent topics in the document, and similarly, edges are generated based on the community memberships of the node pairs. The tensor method is even faster for topic modeling, since the word vocabulary size is typically much smaller than the size of real-world networks. We learn interesting hidden topics in New York Times corpus from UCI bag-of-words data set[1] with around $100,000$ words and $300,000$ documents in about two minutes. We present the important

1. `https://archive.ics.uci.edu/ml/datasets/Bag+of+Words`

words for recovered topics, as well as interpret "bridging" words, which occur in many topics.

Implementations: We present two implementations, viz., a GPU-based implementation which exploits the parallelism of SIMD architectures and a CPU-based implementation for larger datasets, where the GPU memory does not suffice. We discuss various aspects involved such as implicit manipulation of tensors since explicitly forming tensors would be unwieldy for large networks, optimizing for communication bottlenecks in a parallel deployment, the need for sparse matrix and vector operations since real world networks tend to be sparse, and a careful statistical approach to validating the results, when ground truth is available.

1.2 Related work

This paper builds on the recent works of Anandkumar et al (Anandkumar et al., 2012, 2013b) which establishes the correctness of tensor-based approaches for learning MMSB (Airoldi et al., 2008) models and other latent variable models. While, the earlier works provided a theoretical analysis of the method, the current paper considers a careful implementation of the method. Moreover, there are a number of algorithmic improvements in this paper. For instance, while (Anandkumar et al., 2012, 2013b) consider tensor power iterations, based on batch data and deflations performed serially, here, we adopt a stochastic gradient descent approach for tensor decomposition, which provides the flexibility to trade-off sub-sampling with accuracy. Moreover, we use randomized methods for dimensionality reduction in the preprocessing stage of our method which enables us to scale our method to graphs with millions of nodes.

There are other known methods for learning the stochastic block model based on techniques such as spectral clustering (McSherry, 2001) and convex optimization (Chen et al., 2012). However, these methods are not applicable for learning overlapping communities. We note that learning the mixed membership model can be reduced to a matrix factorization problem (Zhang and Yeung, 2012). While collaborative filtering techniques such as (Mnih and Salakhutdinov, 2007; Salakhutdinov and Mnih, 2008) focus on matrix factorization and the prediction accuracy of recommendations on an unseen test set, we recover the underlying latent communities, which helps with the interpretability and the statistical model can be employed for other tasks.

Although there have been other fast implementations for community detection before (Soman and Narang, 2011; Lancichinetti and Fortunato, 2009), these methods are not statistical and do not yield descriptive statistics such as bridging nodes (Nepusz et al., 2008), and cannot perform predictive tasks such as link classification which are the main strengths of the MMSB model. With the implementation of our tensor-based approach, we record huge speed-ups compared to existing approaches for learning the MMSB model.

To the best of our knowledge, while stochastic methods for matrix decomposition have been considered earlier (Oja and Karhunen, 1985; Arora et al., 2012), this is the first work incorporating stochastic optimization for tensor decomposition, and paves the way for further investigation on many theoretical and practical issues. We also note that we never explicitly form or store the subgraph count tensor, of size $O(n^3)$ where n is the number of nodes, in our implementation, but directly manipulate the neighborhood vectors to obtain

tensor decompositions through stochastic updates. This is a crucial departure from other works on tensor decompositions on GPUs (Ballard et al., 2011; Schatz et al., 2013), where the tensor needs to be stored and manipulated directly.

2. Tensor Forms for Topic and Community Models

In this section, we briefly recap the topic and community models, as well as the tensor forms for their exact moments, derived in (Anandkumar et al., 2012, 2013b).

2.1 Topic Modeling

In topic modeling, a document is viewed as a bag of words. Each document has a latent set of topics, and $h = (h_1, h_2, \ldots, h_k)$ represents the proportions of k topics in a given document. Given the topics h, the words are independently drawn and are exchangeable, and hence, the term "bag of words" model. We represent the words in the document by d-dimensional random vectors $x_1, x_2, \ldots x_l \in \mathbb{R}^d$, where x_i are coordinate basis vectors in $\mathbb{R}^d$ and d is the size of the word vocabulary. Conditioned on h, the words in a document satisfy $\mathbb{E}[x_i|h] = \mu h$, where $\mu := [\mu_1, \ldots, \mu_k]$ is the topic-word matrix. And thus μ_j is the topic vector satisfying $\mu_j = \Pr(x_i|h_j)$, $\forall j \in [k]$. Under the Latent Dirichlet Allocation (LDA) topic model (Blei, 2012), h is drawn from a Dirichlet distribution with concentration parameter vector $\alpha = [\alpha_1, \ldots, \alpha_k]$. In other words, for each document u, $h_u \overset{iid}{\sim} \mathrm{Dir}(\alpha)$, $\forall u \in [n]$ with parameter vector $\alpha \in \mathbb{R}^k_+$. We define the Dirichlet concentration (mixing) parameter

$$\alpha_0 := \sum_{i \in [k]} \alpha_i.$$

The Dirichlet distribution allows us to specify the extent of overlap among the topics by controlling for sparsity in topic density function. A larger α_0 results in more overlapped (mixed) topics. A special case of $\alpha_0 = 0$ is the single topic model.

Due to exchangeability, the order of the words does not matter, and it suffices to consider the frequency vector for each document, which counts the number of occurrences of each word in a document. Let $c_t := (c_{1,t}, c_{2,t}, \ldots, c_{d,t}) \in \mathbb{R}^d$ denote the frequency vector for t^{th} document, and let n be the number of documents.

We consider the first three order empirical moments, given by

$$M_1^{\text{Top}} := \frac{1}{n}\sum_{t=1}^{n} c_t \tag{1}$$

$$M_2^{\text{Top}} := \frac{\alpha_0+1}{n}\sum_{t=1}^{n}(c_t \otimes c_t - \text{diag}\,(c_t)) - \alpha_0 M_1^{\text{Top}} \otimes M_1^{\text{Top}} \tag{2}$$

$$\begin{aligned} M_3^{\text{Top}} := {} & \frac{(\alpha_0+1)(\alpha_0+2)}{2n}\sum_{t=1}^{n}\Bigg[c_t \otimes c_t \otimes c_t - \sum_{i=1}^{d}\sum_{j=1}^{d} c_{i,t}c_{j,t}(e_i \otimes e_i \otimes e_j) \\ & - \sum_{i=1}^{d}\sum_{j=1}^{d} c_{i,t}c_{j,t}(e_i \otimes e_j \otimes e_i) - \sum_{i=1}^{d}\sum_{j=1}^{d} c_{i,t}c_{j,t}(e_i \otimes e_j \otimes e_j) + 2\sum_{i=1}^{d} c_{i,t}(e_i \otimes e_i \otimes e_i)\Bigg] \\ & - \frac{\alpha_0(\alpha_0+1)}{2n}\sum_{t=1}^{n}\left(\sum_{i=1}^{d} c_{i,t}(e_i \otimes e_i \otimes M_1^{\text{Top}}) + \sum_{i=1}^{d} c_{i,t}(e_i \otimes M_1^{\text{Top}} \otimes e_i) + \sum_{i=1}^{d} c_{i,t}(M_1^{\text{Top}} \otimes e_i \otimes e_i)\right) \\ & + \alpha_0^2 M_1^{\text{Top}} \otimes M_1^{\text{Top}} \otimes M_1^{\text{Top}}. \end{aligned} \tag{3}$$

We recall Theorem 3.5 of (Anandkumar et al., 2012):

Lemma 1 *The exact moments can be factorized as*

$$\mathbb{E}[M_1^{\text{Top}}] = \sum_{i=1}^{k} \frac{\alpha_i}{\alpha_0}\mu_i \tag{4}$$

$$\mathbb{E}[M_2^{\text{Top}}] = \sum_{i=1}^{k} \frac{\alpha_i}{\alpha_0}\mu_i \otimes \mu_i \tag{5}$$

$$\mathbb{E}[M_3^{\text{Top}}] = \sum_{i=1}^{k} \frac{\alpha_i}{\alpha_0}\mu_i \otimes \mu_i \otimes \mu_i. \tag{6}$$

where $\mu = [\mu_1, \ldots, \mu_k]$ *and* $\mu_i = \Pr(x_t|h=i)$, $\forall t \in [l]$. *In other words,* μ *is the topic-word matrix.*

From the Lemma 1, we observe that the first three moments of a LDA topic model have a simple form involving the topic-word matrix μ and Dirichlet parameters α_i. In (Anandkumar et al., 2012), it is shown that these parameters can be recovered under a weak non-degeneracy assumption. We will employ tensor decomposition techniques to learn the parameters.

2.2 Mixed Membership Model

In the mixed membership stochastic block model (MMSB), introduced by (Airoldi et al., 2008), the edges in a social network are related to the hidden communities of the nodes. A batch tensor decomposition technique for learning MMSB was derived in (Anandkumar et al., 2013b).

Let n denote the number of nodes, k the number of communities and $G \in \mathbb{R}^{n\times n}$ the adjacency matrix of the graph. Each node $i \in [n]$ has an associated community membership

vector $\pi_i \in \mathbb{R}^k$, which is a latent variable, and the vectors are contained in a simplex, i.e.,

$$\sum_{i\in[k]} \pi_u(i) = 1,\ \forall u \in [n]$$

where the notation $[n]$ denotes the set $\{1, \ldots, n\}$. Membership vectors are sampled from the Dirichlet distribution $\pi_u \overset{iid}{\sim} \mathrm{Dir}(\alpha),\ \forall u \in [n]$ with parameter vector $\alpha \in \mathbb{R}^k_+$ where $\alpha_0 := \sum_{i\in[k]} \alpha_i$. As in the topic modeling setting, the Dirichlet distribution allows us to specify the extent of overlap among the communities by controlling for sparsity in community membership vectors. A larger α_0 results in more overlapped (mixed) memberships. A special case of $\alpha_0 = 0$ is the stochastic block model (Anandkumar et al., 2013b).

The *community connectivity matrix* is denoted by $P \in [0,1]^{k\times k}$ where $P(a,b)$ measures the connectivity between communities a and b, $\forall a, b \in [k]$. We model the adjacency matrix entries as either of the two settings given below:

Bernoulli model: This models a network with unweighted edges. It is used for Facebook and DBLP datasets in Section 6 in our experiments.

$$G_{ij} \overset{iid}{\sim} \mathrm{Ber}(\pi_i^\top P \pi_j),\ \forall i, j \in [n].$$

Poisson model (Karrer and Newman, 2011): This models a network with weighted edges. It is used for the Yelp data set in Section 6 to incorporate the review ratings.

$$G_{ij} \overset{iid}{\sim} \mathrm{Poi}(\pi_i^\top P \pi_j),\ \forall i, j \in [n].$$

The tensor decomposition approach involves up to third order moments, computed from the observed network. In order to compute the moments, we partition the nodes randomly into sets X, A, B, C. Let $F_A := \Pi_A^\top P^\top$, $F_B := \Pi_B^\top P^\top$, $F_C := \Pi_C^\top P^\top$ (where P is the community connectivity matrix and Π is the membership matrix) and $\hat{\alpha} := \left(\frac{\alpha_1}{\alpha_0}, \ldots, \frac{\alpha_k}{\alpha_0}\right)$ denote the normalized Dirichlet concentration parameter. We define pairs over Y_1 and Y_2 as $\mathrm{Pairs}(Y_1, Y_2) := G_{X,Y_1}^\top \otimes G_{X,Y_2}^\top$. Define the following matrices

$$Z_B := \mathrm{Pairs}\,(A, C)\,(\mathrm{Pairs}\,(B, C))^\dagger, \tag{7}$$

$$Z_C := \mathrm{Pairs}\,(A, B)\,(\mathrm{Pairs}\,(C, B))^\dagger. \tag{8}$$

We consider the first three empirical moments, given by

$$M_1{}^{\mathrm{Com}} := \frac{1}{n_X} \sum_{x\in X} G_{x,A}^\top \tag{9}$$

$$M_2{}^{\mathrm{Com}} := \frac{\alpha_0 + 1}{n_X} \sum_{x\in X} Z_C G_{x,C}^\top G_{x,B} Z_B^\top - \alpha_0 \left(M_1{}^{\mathrm{Com}} M_1{}^{\mathrm{Com}\,\top}\right) \tag{10}$$

$$\begin{aligned} M_3{}^{\mathrm{Com}} := {} & \frac{(\alpha_0+1)(\alpha_0+2)}{2n_X} \sum_{x\in X} \left[G_{x,A}^\top \otimes Z_B G_{x,B}^\top \otimes Z_C G_{x,C}^\top\right] + \alpha_0^2 M_1{}^{\mathrm{Com}} \otimes M_1{}^{\mathrm{Com}} \otimes M_1{}^{\mathrm{Com}} \\ & - \frac{\alpha_0(\alpha_0+1)}{2n_X} \sum_{x\in X} \Big[G_{x,A}^\top \otimes Z_B G_{x,B}^\top \otimes M_1{}^{\mathrm{Com}} + G_{x,A}^\top \otimes M_1{}^{\mathrm{Com}} \otimes Z_C G_{x,C}^\top \end{aligned} \tag{11}$$

$$+ M_1{}^{\mathrm{Com}} \otimes Z_B G_{x,B}^\top \otimes Z_C G_{x,C}^\top\Big] \tag{12}$$

We now recap Proposition 2.2 of (Anandkumar et al., 2013a) which provides the form of these moments under expectation.

Lemma 2 *The exact moments can be factorized as*

$$\mathbb{E}[{M_1}^{\text{Com}}|\Pi_A, \Pi_B, \Pi_C] := \sum_{i\in[k]} \hat{\alpha}_i (F_A)_i \tag{13}$$

$$\mathbb{E}[{M_2}^{\text{Com}}|\Pi_A, \Pi_B, \Pi_C] := \sum_{i\in[k]} \hat{\alpha}_i (F_A)_i \otimes (F_A)_i \tag{14}$$

$$\mathbb{E}[{M_3}^{\text{Com}}|\Pi_A, \Pi_B, \Pi_C] := \sum_{i\in[k]} \hat{\alpha}_i (F_A)_i \otimes (F_A)_i \otimes (F_A)_i \tag{15}$$

where $\otimes$ *denotes the* Kronecker product *and* $(F_A)_i$ *corresponds to the* i^{th} *column of* F_A*.*

We observe that the moment forms above for the MMSB model have a similar form as the moments of the topic model in the previous section. Thus, we can employ a unified framework for both topic and community modeling involving decomposition of the third order moment tensors M_3^{Top} and M_3^{Com}. Second order moments M_2^{Top} and M_2^{Com} are used for *preprocessing* of the data (i.e., whitening, which is introduced in detail in Section 3.1). For the sake of the simplicity of the notation, in the rest of the paper, we will use M_2 to denote empirical second order moments for both M_2^{Top} in topic modeling setting, and M_2^{Com} in the mixed membership model setting. Similarly, we will use M_3 to denote empirical third order moments for both M_3^{Top} and M_3^{Com}.

3. Learning using Third Order Moment

Our learning algorithm uses up to the third-order moment to estimate the topic word matrix μ or the community membership matrix Π. First, we obtain co-occurrence of triplet words or subgraph counts (implicitly). Then, we perform preprocessing using second order moment M_2. Then we perform tensor decomposition efficiently using *stochastic gradient descent* (Kushner and Yin, 2003) on M_3. We note that, in our implementation of the algorithm on the Graphics Processing Unit (GPU), linear algebraic operations are extremely fast. We also implement our algorithm on the CPU for large datasets which exceed the memory capacity of GPU and use sparse matrix operations which results in large gains in terms of both the memory and the running time requirements. The overall approach is summarized in Algorithm 1.

3.1 Dimensionality Reduction and Whitening

Whitening step utilizes linear algebraic manipulations to make the tensor symmetric and orthogonal (in expectation). Moreover, it leads to dimensionality reduction since it (implicitly) reduces tensor M_3 of size $O(n^3)$ to a tensor of size k^3, where k is the number of communities. Typically we have $k \ll n$. The whitening step also converts the tensor M_3 to a symmetric orthogonal tensor. The whitening matrix $W \in \mathbb{R}^{n_A \times k}$ satisfies $W^\top M_2 W = I$. The idea is that if the bilinear projection of the second order moment onto W results in the identity matrix, then a trilinear projection of the third order moment onto W would

Algorithm 1 Overall approach for learning latent variable models via a moment-based approach.

Input: Observed data: social network graph or document samples.
Output: Learned latent variable model and infer hidden attributes.

1: Estimate the third order moments tensor M_3 (implicitly). The tensor is not formed explicitly as we break down the tensor operations into vector and matrix operations.
2: Whiten the data, via SVD of M_2, to reduce dimensionality via symmetrization and orthogonalization. The third order moments M_3 are whitened as $\mathcal{T}$.
3: Use stochastic gradient descent to estimate spectrum of whitened (implicit) tensor $\mathcal{T}$.
4: Apply post-processing to obtain the topic-word matrix or the community memberships.
5: If ground truth is known, validate the results using various evaluation measures.

result in an orthogonal tensor. We use multilinear operations to get an orthogonal tensor $\mathcal{T} := M_3(W, W, W)$.

The whitening matrix W is computed via truncated $k-$svd of the second order moments.

$$W = U_{M_2} \Sigma_{M_2}^{-1/2},$$

where U_{M_2} and $\Sigma_{M_2} = \mathrm{diag}(\sigma_{M_2,1}, \ldots, \sigma_{M_2,k})$ are the top k singular vectors and singular values of M_2 respectively. We then perform multilinear transformations on the triplet data using the whitening matrix. The whitened data is thus

$$\begin{aligned} y_A^t &:= \left\langle W, c^t \right\rangle, \\ y_B^t &:= \left\langle W, c^t \right\rangle, \\ y_C^t &:= \left\langle W, c^t \right\rangle, \end{aligned}$$

for the topic modeling, where t denotes the index of the documents. Note that y_A^t, y_B^t and $y_C^t \in \mathbb{R}^k$. Implicitly, the whitened tensor is $\mathcal{T} = \frac{1}{n_X} \sum_{t \in X} y_A^t \otimes y_B^t \otimes y_C^t$ and is a $k \times k \times k$ dimension tensor. Since $k \ll n$, the dimensionality reduction is crucial for our speedup.

3.2 Stochastic Tensor Gradient Descent

In (Anandkumar et al., 2013b) and (Anandkumar et al., 2012), the power method with deflation is used for tensor decomposition where the eigenvectors are recovered by iterating over multiple loops in a serial manner. Furthermore, batch data is used in their iterative power method which makes that algorithm slower than its stochastic counterpart. In addition to implementing a stochastic spectral optimization algorithm, we achieve further speed-up by efficiently parallelizing the stochastic updates.

Let $\mathbf{v} = [v_1 | v_2 | \ldots | v_k]$ be the true eigenvectors. Denote the cardinality of the sample set as n_X, i.e., $n_X := |X|$. Now that we have the whitened tensor, we propose the *Stochastic Tensor Gradient Descent* (STGD) algorithm for tensor decomposition. Consider the tensor

$\mathcal{T} \in \mathbb{R}^{k\times k\times k}$ using whitened samples, i.e.,

$$\mathcal{T} = \sum_{t\in X} \mathcal{T}^t = \frac{(\alpha_0+1)(\alpha_0+2)}{2n_X} \sum_{t\in X} y_A^t \otimes y_B^t \otimes y_C^t$$
$$- \frac{\alpha_0(\alpha_0+1)}{2n_X} \sum_{t\in X} \left[y_A^t \otimes y_B^t \otimes \bar{y}_C + y_A^t \otimes \bar{y}_B \otimes y_C^t + \bar{y}_A \otimes y_B^t \otimes y_C^t \right] + \alpha_0^2 \bar{y}_A \otimes \bar{y}_B \otimes \bar{y}_C,$$

where $t \in X$ and denotes the index of the online data and $\bar{y}_A$, $\bar{y}_B$, and $\bar{y}_C$ denote the mean of the whitened data. Our goal is to find a symmetric CP decomposition of the whitened tensor.

Definition 3 *Our optimization problem is given by*

$$\arg \min_{\mathbf{v}:\|v_i\|_F^2=1} \Big\{ \big\| \sum_{i\in[k]} \otimes^3 v_i - \sum_{t\in X} \mathcal{T}^t \big\|_F^2 + \theta \| \sum_{i\in[k]} \otimes^3 v_i \|_F^2 \Big\},$$

where v_i are the unknown components to be estimated, and $\theta > 0$ is some fixed parameter.

In order to encourage orthogonality between eigenvectors, we have the extra term as $\theta\|\sum_{i\in[k]} \otimes^3 v_i\|_F^2$. Since $\|\sum_{t\in X} \mathcal{T}^t\|_F^2$ is a constant, the above minimization is the same as minimizing a loss function $L(\mathbf{v}) := \frac{1}{n_X} \sum_t L^t(\mathbf{v})$, where $L^t(\mathbf{v})$ is the loss function evaluated at node $t \in X$, and is given by

$$L^t(\mathbf{v}) := \frac{1+\theta}{2} \big\| \sum_{i\in[k]} \otimes^3 v_i \big\|_F^2 - \big\langle \sum_{i\in[k]} \otimes^3 v_i, \mathcal{T}^t \big\rangle \tag{16}$$

The loss function has two terms, viz., the term $\|\sum_{i\in[k]} \otimes^3 v_i\|_F^2$, which can be interpreted as the orthogonality cost, which we need to minimize, and the second term $\langle \sum_{i\in[k]} \otimes^3 v_i, \mathcal{T}^t \rangle$, which can be viewed as the correlation reward to be maximized. The parameter θ provides additional flexibility for tuning between the two terms.

Let $\Phi^t := \left[\phi_1^t | \phi_2^t | \dots | \phi_k^t\right]$ denote the estimation of the eigenvectors using the whitened data point t, where $\phi_i^t \in \mathbb{R}^k,\ i \in [k]$. Taking the derivative of the loss function leads us to the iterative update equation for the stochastic gradient descent which is

$$\phi_i^{t+1} \leftarrow \phi_i^t - \beta^t \frac{\partial L^t}{\partial v_i}\bigg|_{\phi_i^t},\ \forall i \in [k]$$

where β^t is the learning rate. Computing the derivative of the loss function and substituting the result leads to the following lemma.

Lemma 4 *The stochastic updates for the eigenvectors are given by*

$$\phi_i^{t+1} \leftarrow \phi_i^t - \frac{1+\theta}{2}\beta^t \sum_{j=1}^k \left[\left\langle \phi_j^t, \phi_i^t \right\rangle^2 \phi_j^t \right] + \beta^t \frac{(\alpha_0+1)(\alpha_0+2)}{2} \left\langle \phi_i^t, y_A^t \right\rangle \left\langle \phi_i^t, y_B^t \right\rangle y_C^t + \beta^t \alpha_0^2 \left\langle \phi_i^t, \bar{y}_A \right\rangle \left\langle \phi_i^t, \bar{y}_B^t \right\rangle \bar{y}_C$$
$$- \beta^t \frac{\alpha_0(\alpha_0+1)}{2} \left\langle \phi_i^t, y_A^t \right\rangle \left\langle \phi_i^t, y_B^t \right\rangle \bar{y}_C - \beta^t \frac{\alpha_0(\alpha_0+1)}{2} \left\langle \phi_i^t, y_A^t \right\rangle \left\langle \phi_i^t, \bar{y}_B \right\rangle y_C - \beta^t \frac{\alpha_0(\alpha_0+1)}{2} \left\langle \phi_i^t, \bar{y}_A \right\rangle \left\langle \phi_i^t, y_B^t \right\rangle y_C, \tag{17}$$

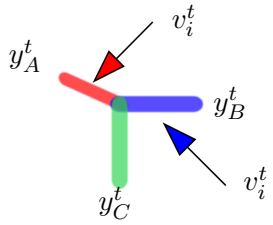


Figure 1: Schematic representation of the stochastic updates for the spectral estimation. Note the we never form the tensor explicitly, since the gradient involves vector products by collapsing two modes, as shown in Equation 17.

In Equation (17), all our tensor operations are in terms of efficient sample vector inner products, and no tensor is explicitly formed. The multilinear operations are shown in Figure 1. We choose $\theta = 1$ in our experiments to ensure that there is sufficient penalty for non-orthogonality, which prevents us from obtaining degenerate solutions.

After learning the decomposition of the third order moment, we perform post-processing to estimate $\widehat{\Pi}$.

3.3 Post-processing

Eigenvalues $\Lambda := [\lambda_1, \lambda_2, \ldots, \lambda_k]$ are estimated as the norm of the eigenvectors $\lambda_i = \|\phi_i\|^3$.

Lemma 5 *After we obtain Λ and Φ, the estimate for the topic-word matrix is given by*

$$\hat{\mu} = W^{\top\dagger}\Phi,$$

and in the community setting, the community membership matrix is given by

$$\hat{\Pi}_{A^c} = \mathrm{diag}(\gamma)^{1/3}\,\mathrm{diag}(\Lambda)^{-1}\Phi^\top \hat{W}^\top G_{A,A^c}.$$

where $A^c := X \cup B \cup C$. Similarly, we estimate $\hat{\Pi}_A$ by exchanging the roles of X and A. Next, we obtain the Dirichlet distribution parameters

$$\hat{\alpha}_i = \gamma^2 \lambda_i^{-2}, \forall i \in [k].$$

where γ^2 is chosen such that we have normalization $\sum_{i\in[k]} \hat{\alpha}_i := \sum_{i\in[k]} \frac{\alpha_i}{\alpha_0} = 1$.

Thus, we perform STGD method to estimate the eigenvectors and eigenvalues of the whitened tensor, and then use these to estimate the topic word matrix μ and community membership matrix $\widehat{\Pi}$ by thresholding.

4. Implementation Details

4.1 Symmetrization Step to Compute M_2

Note that for the topic model, the second order moment M_2 can be computed easily from the word-frequency vector. On the other hand, for the community setting, computing M_2 requires additional linear algebraic operations. It requires computation of matrices Z_B

and Z_C in equation (7). This requires computation of pseudo-inverses of "Pairs" matrices. Now, note that pseudo-inverse of (Pairs (B, C)) in Equation (7) can be computed using rank k-SVD:

$$\text{k-SVD}\left(\text{Pairs}\left(B, C\right)\right) = U_B(:, 1:k)\Sigma_{BC}(1:k)V_C(:, 1:k)^\top.$$

We exploit the low rank property to have efficient running times and storage. We first implement the k-SVD of Pairs, given by $G_{X,C}^\top G_{X,B}$. Then the order in which the matrix products are carried out plays a significant role in terms of both memory and speed. Note that Z_C involves the multiplication of a sequence of matrices of sizes $\mathbb{R}^{n_A \times n_B}$, $\mathbb{R}^{n_B \times k}$, $\mathbb{R}^{k \times k}$, $\mathbb{R}^{k \times n_C}$, $G_{x,C}^\top G_{x,B}$ involves products of sizes $\mathbb{R}^{n_C \times k}$, $\mathbb{R}^{k \times k}$, $\mathbb{R}^{k \times n_B}$, and Z_B involving products of sizes $\mathbb{R}^{n_A \times n_C}$, $\mathbb{R}^{n_C \times k}$, $\mathbb{R}^{k \times k}$, $\mathbb{R}^{k \times n_B}$. While performing these products, we avoid products of sizes $\mathbb{R}^{O(n) \times O(n)}$ and $\mathbb{R}^{O(n) \times O(n)}$. This allows us to have efficient storage requirements. Such manipulations are represented in Figure 2.

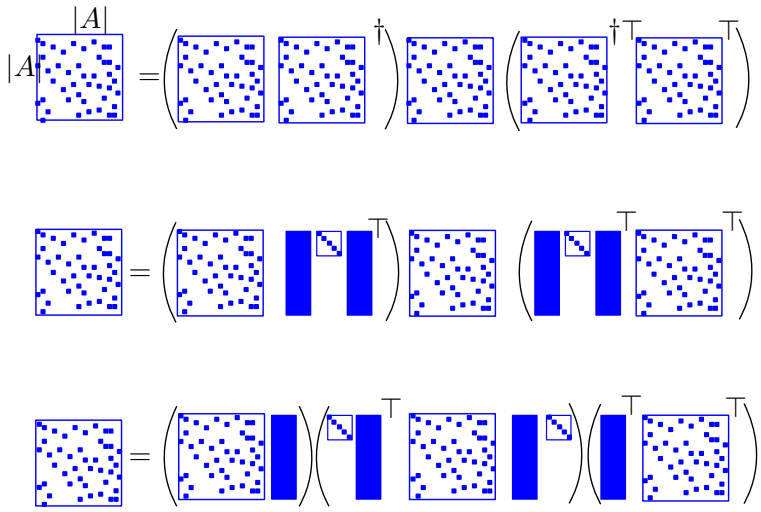


Figure 2: By performing the matrix multiplications in an efficient order (Equation (10)), we avoid products involving $O(n) \times O(n)$ objects. Instead, we use objects of size $O(n) \times k$ which improves the speed, since $k \ll n$. Equation (10) is equivalent to $M_2 = \left(\text{Pairs}_{A,B}\,\text{Pairs}_{C,B}^\dagger\right)\,\text{Pairs}_{C,B}\,\left(\text{Pairs}_{B,C}^\dagger\right)^\top\,\text{Pairs}_{A,C}^\top$ $-$shift, where the shift $= \frac{\alpha_0}{\alpha_0+1}\left(M_1 M_1{}^\top - \text{diag}\left(M_1 M_1{}^\top\right)\right)$. We do not explicitly calculate the pseudoinverse but maintain the low rank matrix decomposition form.

We then orthogonalize the third order moments to reduce the dimension of its modes to k. We perform linear transformations on the data corresponding to the partitions A, B and C using the whitening matrix. The whitened data is thus $y_A^t := \left\langle W, G_{t,A}^\top \right\rangle$, $y_B^t := \left\langle W, Z_B G_{t,B}^\top \right\rangle$, and $y_C^t := \left\langle W, Z_C G_{t,C}^\top \right\rangle$, where $t \in X$ and denotes the index of the online data. Since $k \ll n$, the dimensionality reduction is crucial for our speedup.

4.2 Efficient Randomized SVD Computations

When we consider very large-scale data, the whitening matrix is a bottleneck to handle when we aim for fast running times. We obtain the low rank approximation of matrices using random projections. In the CPU implementation, we use *tall-thin SVD* (on a sparse matrix) via the Lanczos algorithm after the projection and in the GPU implementation, we use *tall-thin QR*. We give the overview of these methods below. Again, we use graph community membership model without loss of generality.

Randomized low rank approximation: From (Gittens and Mahoney, 2013), for the k-rank positive semi-definite matrix $M_2 \in \mathbb{R}^{n_A \times n_A}$ with $n_A \gg k$, we can perform random projection to reduce dimensionality. More precisely, if we have a random matrix $S \in \mathbb{R}^{n_A \times \tilde{k}}$ with unit norm (rotation matrix), we project M_2 onto this random matrix to get $\mathbb{R}^{n \times \tilde{k}}$ tall-thin matrix. Note that we choose $\tilde{k} = 2k$ in our implementation. We will obtain lower dimension approximation of M_2 in $\mathbb{R}^{\tilde{k} \times \tilde{k}}$. Here we emphasize that $S \in \mathbb{R}^{n \times \tilde{k}}$ is a random matrix for dense M_2. However for sparse M_2, $S \in \{0,1\}^{n \times \tilde{k}}$ is a column selection matrix with random sign for each entry.

After the projection, one approach we use is SVD on this tall-thin ($\mathbb{R}^{n \times \tilde{k}}$) matrix. Define $O := M_2 S \in \mathbb{R}^{n \times \tilde{k}}$ and $\Omega := S^\top M_2 S \in \mathbb{R}^{\tilde{k} \times \tilde{k}}$. A low rank approximation of M_2 is given by $O \Omega^\dagger O^\top$ (Gittens and Mahoney, 2013). Recall that the definition of a whitening matrix W is that $W^\top M_2 W = I$. We can obtain the whitening matrix of M_2 without directly doing a SVD on $M_2 \in \mathbb{R}^{n_A \times n_A}$.

Tall-thin SVD: This is used in the CPU implementation. The whitening matrix can be obtained by

$$W \approx (O^\dagger)^\top (\Omega^{\frac{1}{2}})^\top. \tag{18}$$

The pseudo code for computing the whitening matrix W using tall-thin SVD is given in Algorithm 2. Therefore, we only need to compute SVD of a tall-thin matrix $O \in \mathbb{R}^{n_A \times \tilde{k}}$.

Algorithm 2 Randomized Tall-thin SVD

Input: Second moment matrix M_2.

Output: Whitening matrix W.

1: Generate random matrix $S \in \mathbb{R}^{n \times \tilde{k}}$ if M_2 is dense.
2: Generate column selection matrix with random sign $S \in \{0,1\}^{n \times \tilde{k}}$ if M_2 is sparse.
3: $O = M_2 S \in \mathbb{R}^{n \times \tilde{k}}$
4: $[U_O, L_O, V_O] =$SVD(O)
5: $\Omega = S^\top O \in \mathbb{R}^{\tilde{k} \times \tilde{k}}$
6: $[U_\Omega, L_\Omega, V_\Omega] =SVD(\Omega)$
7: $W = U_O L_O^{-1} V_O^\top V_\Omega L_\Omega^{\frac{1}{2}} U_\Omega^\top$

Note that $\Omega \in \mathbb{R}^{\tilde{k} \times \tilde{k}}$, its square-root is easy to compute. Similarly, pseudoinverses can also be obtained without directly doing SVD. For instance, the pseudoinverse of the Pairs (B, C) matrix is given by

$$(\text{Pairs}\,(B,C))^\dagger = (J^\dagger)^\top \Psi J^\dagger,$$

where $\Psi = S^\top (\text{Pairs}\,(B,C))\, S$ and $J = (\text{Pairs}\,(B,C))\, S$. The pseudo code for computing pseudoinverses is given in Algorithm 3.

Algorithm 3 Randomized Pseudoinverse

Input: Pairs matrix $\text{Pairs}\,(B, C)$.

Output: Pseudoinverse of the pairs matrix $(\text{Pairs}\,(B, C))^\dagger$.

1: Generate random matrix $S \in \mathbb{R}^{n,k}$ if M_2 is dense.
2: Generate column selection matrix with random sign $S \in \{0, 1\}^{n \times k}$ if M_2 is sparse.
3: $J = (\text{Pairs}\,(B, C))\, S$
4: $\Psi = S^\top J$
5: $[U_J, L_J, V_J]$ =SVD(J)
6: $(\text{Pairs}\,(B, C))^\dagger = U_J L_J^{-1} V_J^\top \Psi V_J L_J^{-1} U_J^\top$

The sparse representation of the data allows for scalability on a single machine to datasets having millions of nodes. Although the GPU has SIMD architecture which makes parallelization efficient, it lacks advanced libraries with sparse SVD operations and out-of-GPU-core implementations. We therefore implement the sparse format on CPU for sparse datasets. We implement our algorithm using random projection for efficient dimensionality reduction (Clarkson and Woodruff, 2012) along with the sparse matrix operations available in the Eigen toolkit[2], and we use the SVDLIBC (Berry et al., 2002) library to compute sparse SVD via the Lanczos algorithm. Theoretically, the Lanczos algorithm (Golub and Van Loan, 2013) on a $n \times n$ matrix takes around $(2d + 8)n$ flops for a single step where d is the average number of non-zero entries per row.

Tall-thin QR: This is used in the GPU implementation due to the lack of library to do sparse tall-thin SVD. The difference is that we instead implement a tall-thin QR on O, therefore the whitening matrix is obtained as

$$W \approx Q(R^\dagger)^\top(\Omega^{\frac{1}{2}})^\top.$$

The main bottleneck for our GPU implementation is device storage, since GPU memory is highly limited and not expandable. Random projections help in reducing the dimensionality from $O(n \times n)$ to $O(n \times k)$ and hence, this fits the data in the GPU memory better. Consequently, after the whitening step, we project the data into k-dimensional space. Therefore, the STGD step is dependent only on k, and hence can be fit in the GPU memory. So, the main bottleneck is computation of large SVDs. In order to support larger datasets such as the DBLP data set which exceed the GPU memory capacity, we extend our implementation with out-of-GPU-core matrix operations and the Nystrom method (Gittens and Mahoney, 2013) for the whitening matrix computation and the pseudoinverse computation in the pre-processing module.

4.3 Stochastic updates

STGD can potentially be the most computationally intensive task if carried out naively since the storage and manipulation of a $O(n^3)$-sized tensor makes the method not scalable. However we overcome this problem since we never form the tensor explicitly; instead, we collapse the tensor modes implicitly as shown in Figure 1. We gain large speed up by optimizing the implementation of STGD.To implement the tensor operations efficiently we

2. `http://eigen.tuxfamily.org/index.php?title=Main_Page`

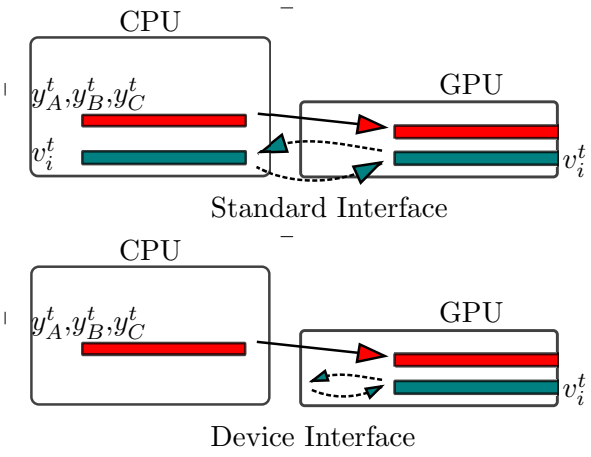


Figure 3: Data transfers in the standard and device interfaces of the GPU implementation.

convert them into matrix and vector operations so that they are implemented using BLAS routines. We obtain whitened vectors y_A, y_B and y_C and manipulate these vectors efficiently to obtain tensor eigenvector updates using the gradient scaled by a suitable learning rate.

Efficient STGD via stacked vector operations: We convert the BLAS II into BLAS III operations by stacking the vectors to form matrices, leading to more efficient operations. Although the updating equation for the stochastic gradient update is presented serially in Equation (17), we can update the k eigenvectors simultaneously in parallel. The basic idea is to stack the k eigenvectors $\phi_i \in \mathbb{R}^k$ into a matrix $\mathbf{\Phi}$, then using the internal parallelism designed for BLAS III operations.

Overall, the STGD step involves $1+k+i(2+3k)$ BLAS II over $\mathbb{R}^k$ vectors, 7N BLAS III over $\mathbb{R}^{k\times k}$ matrices and 2 QR operations over $\mathbb{R}^{k\times k}$ matrices, where i denotes the number of iterations. We provide a count of BLAS operations for various steps in Table 1.

Module	BLAS I	BLAS II	BLAS III	SVD	QR
Pre	0	8	19	3	0
STGD	0	Nk	$7N$	0	2
Post	0	0	7	0	0

Table 1: Linear algebraic operation counts: N denotes the number of iterations for STGD and k, the number of communities.

Reducing communication in GPU implementation: In STGD, note that the storage needed for the iterative part does not depend on the number of nodes in the data set, rather, it depends on the parameter k, i.e., the number of communities to be estimated, since whitening performed before STGD leads to dimensionality reduction. This makes it suitable for storing the required buffers in the GPU memory, and using the CULA device interface for the BLAS operations. In Figure 3, we illustrate the data transfer involved in the GPU standard and device interface codes. While the standard interface involves data

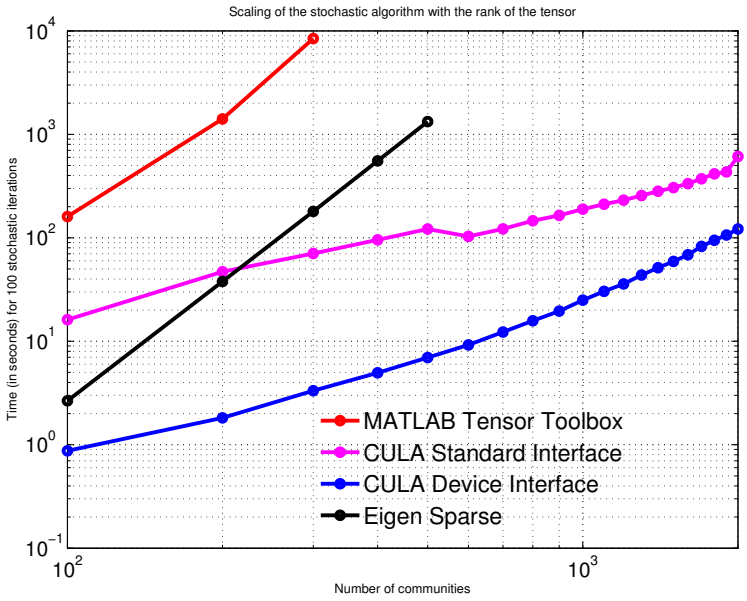


Figure 4: Comparison of the running time for STGD under different k for 100 iterations.

transfer (including whitened neighborhood vectors and the eigenvectors) at each stochastic iteration between the CPU memory and the GPU memory, the device interface involves allocating and retaining the eigenvectors at each stochastic iteration which in turn speeds up the spectral estimation.

We compare the running time of the CULA device code with the MATLAB code (using the tensor toolbox (Bader et al., 2012)), CULA standard code and Eigen sparse code in Figure 4. As expected, the GPU implementations of matrix operations are much faster and scale much better than the CPU implementations. Among the CPU codes, we notice that sparsity and optimization offered by the Eigen toolkit gives us huge gains. We obtain orders of magnitude of speed up for the GPU device code as we place the buffers in the GPU memory and transfer minimal amount of data involving the whitened vectors only once at the beginning of each iteration. The running time for the CULA standard code is more than the device code because of the CPU-GPU data transfer overhead. For the same reason, the sparse CPU implementation, by avoiding the data transfer overhead, performs better than the GPU standard code for very small number of communities. We note that there is no performance degradation due to the parallelization of the matrix operations. After whitening, the STGD requires the most code design and optimization effort, and so we convert that into BLAS-like routines.

4.4 Computational Complexity

We partition the execution of our algorithm into three main modules namely, pre-processing, STGD and post-processing, whose various matrix operation counts are listed above in Table 1.

Module	Time	Space
Pre-processing (Matrix Multiplication)	$O\left(\max(nsk/c, \log s)\right)$	$O\left(\max(s^2, sk)\right)$
Pre-processing (CPU SVD)	$O\left(\max(nsk/c, \log s) + \max(k^2/c, k)\right)$	$O(sk)$
Pre-processing (GPU QR)	$O\left(\max(sk^2/c, \log s) + \max(sk^2/c, \log k)\right)$	$O(sk)$
Pre-processing(short-thin SVD)	$O\left(\max(k^3/c, \log k) + \max(k^2/c, k)\right)$	$O(k^2)$
STGD	$O\left(\max(k^3/c, \log k)\right)$	$O(k^2)$
Post-processing	$O\left(\max(nsk/c, \log s)\right)$	$O(nk)$

Table 2: The time and space complexity (number of compute cores required) of our algorithm. Note that $k \ll n$, s is the average degree of a node (or equivalently, the average number of non-zeros per row/column in the adjacency sub-matrix); note that the STGD time is per iteration time. We denote the number of cores as c - the time-space trade-off depends on this parameter.

The theoretical asymptotic complexity of our method is summarized in Table 2 and is best addressed by considering the parallel model of computation (JáJá, 1992), i.e., wherein a number of processors or compute cores are operating on the data simultaneously in parallel. This is justified considering that we implement our method on GPUs and matrix products are embarrassingly parallel. Note that this is different from serial computational complexity. We now break down the entries in Table 2. First, we recall a basic lemma regarding the lower bound on the time complexity for parallel addition along with the required number of cores to achieve a speed-up.

Lemma 6 *(JáJá, 1992) Addition of s numbers in serial takes $O(s)$ time; with $\Omega(s/\log s)$ cores, this can be improved to $O(\log s)$ time in the best case.*

Essentially, this speed-up is achieved by recursively adding pairs of numbers in parallel.

Lemma 7 *(JáJá, 1992) Consider $M \in \mathbb{R}^{p\times q}$ and $N \in \mathbb{R}^{q\times r}$ with s non-zeros per row/column. Naive serial matrix multiplication requires $O(psr)$ time; with $\Omega(psr/\log s)$ cores, this can be improved to $O(\log s)$ time in the best case.*

Lemma 7 follows by simply parallelizing the sparse inner products and applying Lemma 6 for the addition in the inner products. Note that, this can be generalized to the fact that given c cores, the multiplication can be performed in $O(\max(psr/c, \log s))$ running time.

4.4.1 Pre-processing

Random projection: In preprocessing, given c compute cores, we first do random projection using matrix multiplication. We multiply an $O(n) \times O(n)$ matrix M_2 with an $O(n) \times O(k)$ random matrix S. Therefore, this requires $O(nsk)$ serial operations, where s is the number of non-zero elements per row/column of M_2. Using Lemma 7, given $c = \frac{nsk}{\log s}$ cores, we could achieve $O(\log s)$ computational complexity. However, the parallel computational complexity is not further reduced with more than $\frac{nsk}{\log s}$ cores.

After the multiplication, we use *tall-thin SVD* for CPU implementation, and *tall-thin QR* for GPU implementation.

Tall-thin SVD: We perform Lanczos SVD on the tall-thin sparse $O(n) \times O(k)$ matrix, which involves a tri-diagonalization followed with the QR on the tri-diagonal matrix. Given $c = \frac{nsk}{\log s}$ cores, the computational complexity of the tri-diagonalization is $O(\log s)$. We then do QR on the tridiagonal matrix which is as cheap as $O(k^2)$ serially. Each orthogonalization requires $O(k)$ inner products of constant entry vectors, and there are $O(k)$ such orthogonalizations to be done. Therefore given $O(k)$ cores, the complexity is $O(k)$. More cores does not help since the degree of parallelism is k.

Tall-thin QR: Alternatively, we perform QR in the GPU implementation which takes $O(sk^2)$. To arrive at the complexity of obtaining Q, we analyze the Gram-Schmidt orthonormalization procedure under sparsity and parallelism conditions. Consider a serial Gram-Schmidt on k columns (which are s-dense) of $O(n) \times O(k)$ matrix. For each of the columns 2 to k, we perform projection on the previously computed components and subtract it. Both inner product and subtraction operations are on the s-dense columns and there are $O(s)$ operations which are done $O(k^2)$ times serially. The last step is the normalization of k s-dense vectors with is an $O(sk)$ operation. This leads to a serial complexity of $O(sk^2 + sk) = O(sk^2)$. Using this, we may obtain the parallel complexity in different regimes of the number of cores as follows.

Parallelism for inner products : For each component i, we need $i-1$ projections on previous components which can be parallel. Each projection involves scaling and inner product operations on a pair of s-dense vectors. Using Lemma 6, projection for component i can be performed in $O(\max(\frac{sk}{c}, \log s))$ time. $O(\log s)$ complexity is obtained using $O(sk/\log s)$ cores.

Parallelism for subtractions: For each component i, we need $i-1$ subtractions on a s-dense vector after the projection. Serially the subtraction requires $O(sk)$ operations, and this can be reduced to $O(\log k)$ with $O(sk/\log k)$ cores in the best case. The complexity is $O(\max(\frac{sk}{c}, \log k))$.

Combing the inner products and subtractions, the complexity is $O\left(\max(\frac{sk}{c}, \log s) + \max(\frac{sk}{c}, \log k)\right)$ for component i. There are k components in total, which can not be parallel. In total, the complexity for the parallel QR is $O\left(\max(\frac{sk^2}{c}, \log s) + \max(\frac{sk^2}{c}, \log k)\right)$.

Short-thin SVD: SVD of the smaller $O(\mathbb{R}^{k\times k})$ matrix time requires $O(k^3)$ computations in serially. We note that this is the bottleneck for the computational complexity, but we emphasize that k is sufficiently small in many applications. Furthermore, this k^3 complexity can be reduced by using distributed SVD algorithms e.g. (Kannan et al., 2014; Feldman et al., 2013). An analysis with respect to Lanczos parallel SVD is similar with the discussion in the Tall-thin SVD paragraph. The complexity is $O(\max(k^3/c, \log k) + \max(k^2/c, k))$. In the best case, the complexity is reduced to $O(\log k + k)$.

The serial time complexity of SVD is $O(n^2k)$ but with randomized dimensionality reduction (Gittens and Mahoney, 2013) and parallelization (Constantine and Gleich, 2011), this is significantly reduced.

4.4.2 STGD

In STGD, we perform implicit stochastic updates, consisting of a constant number of matrix-matrix and matrix-vector products, on the set of eigenvectors and whitened samples which is of size $k \times k$. When $c \in [1, k^3/\log k]$, we obtain a running time of $O(k^3/c)$ for computing

inner products in parallel with c compute cores since each core can perform an inner product to compute an element in the resulting matrix independent of other cores in linear time. For $c \in (k^3/\log k, \infty]$, using Lemma 6, we obtain a running time of $O(\log k)$. Note that the STGD time complexity is calculated per iteration.

4.4.3 Post-processing

Finally, post-processing consists of sparse matrix products as well. Similar to pre-processing, this consists of multiplications involving the sparse matrices. Given s number of non-zeros per column of an $O(n) \times O(k)$ matrix, the effective number of elements reduces to $O(sk)$. Hence, given $c \in [1, nks/\log s]$ cores, we need $O(nsk/c)$ time to perform the inner products for each entry of the resultant matrix. For $c \in (nks/\log s, \infty]$, using Lemma 6, we obtain a running time of $O(\log s)$.

Note that nk^2 is the complexity of computing the exact SVD and we reduce it to $O(k)$ when there are sufficient cores available. This is meant for the setting where k is small. This k^3 complexity of SVD on $O(k \times k)$ matrix can be reduced to $O(k)$ using distributed SVD algorithms e.g. (Kannan et al., 2014; Feldman et al., 2013). We note that the variational inference algorithm complexity, by Gopalan and Blei (Gopalan and Blei, 2013), is $O(mk)$ for each iteration, where m denotes the number of edges in the graph, and $n < m < n^2$. In the regime that $n \gg k$, our algorithm is more efficient. Moreover, a big difference is in the scaling with respect to the size of the network and ease of parallelization of our method compared to variational one.

5. Validation methods

5.1 p-value testing:

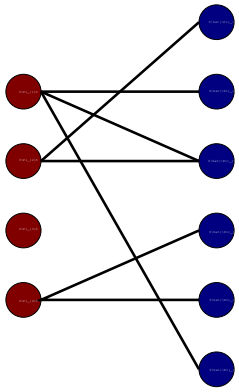


Figure 5: Bipartite graph $G_{\{\mathsf{P}_{\text{val}}\}}$ induced by p-value testing. Edges represent statistically significant relationships between ground truth and estimated communities.

We recover the estimated community membership matrix $\widehat{\Pi} \in \mathbb{R}^{\widehat{k} \times n}$, where $\widehat{k}$ is the number of communities specified to our method. Recall that the true community membership matrix is Π, and we consider datasets where ground truth is available. Let i-th row of

$\widehat{\Pi}$ be denoted by $\widehat{\Pi}_i$. Our community detection method is unsupervised, which inevitably results in row permutations between Π and $\widehat{\Pi}$ and $\widehat{k}$ may not be the same as k. To validate the results, we need to find a good match between the rows of $\widehat{\Pi}$ and Π. We use the notion of p-values to test for statistically significant dependencies among a set of random variables. The p-value denotes the probability of not rejecting the null hypothesis that the random variables under consideration are independent and we use the Student's[3] t-test statistic (Fadem, 2012) to compute the p-value. We use multiple hypothesis testing for different pairs of estimated and ground-truth communities $\widehat{\Pi}_i, \Pi_j$ and adjust the p-values to ensure a small enough false discovery rate (FDR) (Strimmer, 2008).

The test statistic used for the p-value testing of the estimated communities is

$$T_{ij} := \frac{\rho\left(\widehat{\Pi}_i, \Pi_j\right)\sqrt{n-2}}{\sqrt{1-\rho\left(\widehat{\Pi}_i, \Pi_j\right)^2}}.$$

The right p-value is obtained via the probability of obtaining a value (say t_{ij}) greater than the test statistic T_{ij}, and it is defined as

$$\mathsf{P}_{\text{val}}(\Pi_i, \widehat{\Pi}_j) := 1 - \mathbb{P}\left(t_{ij} > T_{ij}\right).$$

Note that T_{ij} has Student's t-distribution with degree of freedom $n-2$ (i.e. $T_{ij} \sim t_{n-2}$). Thus, we obtain the right p-value[4].

In this way, we compute the $\mathbf{P}_{\text{val}}$ matrix as

$$\mathbf{P}_{\text{val}}(i,j) := \mathsf{P}_{\text{val}}\left[\widehat{\Pi}_i, \Pi_j\right], \forall i \in [k] \text{ and } j \in [\widehat{k}].$$

5.2 Evaluation metrics

Recovery ratio: Validating the results requires a matching of the true membership Π with estimated membership $\widehat{\Pi}$. Let $\mathsf{P}_{\text{val}}(\Pi_i, \widehat{\Pi}_j)$ denote the right p-value under the null hypothesis that Π_i and $\widehat{\Pi}_j$ are statistically independent. We use the p-value test to find out pairs $\Pi_i, \widehat{\Pi}_j$ which pass a specified p-value threshold, and we denote such pairs using a bipartite graph $G_{\{\mathsf{P}_{\text{val}}\}}$. Thus, $G_{\{\mathsf{P}_{\text{val}}\}}$ is defined as

$$G_{\{\mathsf{P}_{\text{val}}\}} := \left(\left\{V^{(1)}_{\{\mathsf{P}_{\text{val}}\}}, V^{(2)}_{\{\mathsf{P}_{\text{val}}\}}\right\}, E_{\{\mathsf{P}_{\text{val}}\}}\right),$$

where the nodes in the two node sets are

$$V^{(1)}_{\{\mathsf{P}_{\text{val}}\}} = \{\Pi_1, \ldots, \Pi_k\},$$
$$V^{(2)}_{\{\mathsf{P}_{\text{val}}\}} = \left\{\widehat{\Pi}_1, \ldots, \widehat{\Pi}_{\widehat{k}}\right\}$$

3. Note that Student's t-test is robust to the presence of unequal variances when the sample sizes of the two are equal which is true in our setting.
4. The right p-value accounts for the fact that when two communities are anti-correlated they are not paired up. Hence note that in the special case of block model in which the estimated communities are just permuted version of the ground truth communities, the pairing results in a perfect matching accurately.

and the edges of $G_{\{\mathsf{P}_{\text{val}}\}}$ satisfy

$$(i,j) \in E_{\{\mathsf{P}_{\text{val}}\}} \text{ s.t. } \mathsf{P}_{\text{val}}\left[\widehat{\Pi}_i, \Pi_j\right] \leq 0.01.$$

A simple example is shown in Figure 5, in which Π_2 has statistically significant dependence with $\widehat{\Pi}_1$, i.e., the probability of not rejecting the null hypothesis is small (recall that null hypothesis is that they are independent). If no estimated membership vector has a significant overlap with Π_3, then Π_3 is not recovered. There can also be multiple pairings such as for Π_1 and $\{\widehat{\Pi}_2, \widehat{\Pi}_3, \widehat{\Pi}_6\}$. The p-value test between Π_1 and $\{\widehat{\Pi}_2, \widehat{\Pi}_3, \widehat{\Pi}_6\}$ indicates that probability of not rejecting the null hypothesis is small, i.e., they are independent. We use 0.01 as the threshold. The same holds for Π_2 and $\{\widehat{\Pi}_1\}$ and for Π_4 and $\{\widehat{\Pi}_4, \widehat{\Pi}_5\}$. There can be a perfect one to one matching like for Π_2 and $\widehat{\Pi}_1$ as well as a multiple matching such as for Π_1 and $\{\widehat{\Pi}_2, \widehat{\Pi}_3, \widehat{\Pi}_6\}$. Or another multiple matching such as for $\{\Pi_1, \Pi_2\}$ and $\widehat{\Pi}_3$.

Let Degree_i denote the degree of ground truth community $i \in [k]$ in $G_{\{\mathsf{P}_{\text{val}}\}}$, we define the recovery ratio as follows.

Definition 8 *The* recovery ratio *is defined as*

$$\mathcal{R} := \frac{1}{k} \sum_i \mathbb{I}\left\{\text{Degree}_i > 0\right\}, \quad i \in [k]$$

where $\mathbb{I}(x)$ is the indicator function whose value equals one if x is true.

The perfect case is that all the memberships have at least one significant overlapping estimated membership, giving a recovery ratio of 100%. *Error function:* For performance analysis of our learning algorithm, we use an error function given as follows:

Definition 9 *The average error function is defined as*

$$\mathcal{E} := \frac{1}{k} \sum_{(i,j) \in E_{\{\mathsf{P}_{val}\}}} \left\{ \frac{1}{n} \sum_{x \in |X|} \left| \widehat{\Pi}_i(x) - \Pi_j(x) \right| \right\},$$

where $E_{\{\mathsf{P}_{val}\}}$ denotes the set of edges based on thresholding of the p-values.

The error function incorporates two aspects, namely the l_1 norm error between each estimated community and the corresponding paired ground truth community, and the error induced by false pairings between the estimated and ground-truth communities through p-value testing. For the former l_1 norm error, we normalize with n which is reasonable and results in the range of the error in $[0, 1]$. For the latter, we define the average error function as the summation of all paired memberships errors divided by the true number of communities k. In this way we penalize falsely discovered pairings by summing them up. Our error function can be greater than 1 if there are too many falsely discovered pairings through p-value testing (which can be as large as $k \times \widehat{k}$).

Bridgeness: Bridgeness in overlapping communities is an interesting measure to evaluate. A bridge is defined as a vertex that crosses structural holes between discrete groups of

Hardware / software	Version
CPU	Dual 8-core Xeon @ 2.0GHz
Memory	64GB DDR3
GPU	Nvidia Quadro K5000
CUDA Cores	1536
Global memory	4GB GDDR5
CentOS	Release 6.4 (Final)
GCC	4.4.7
CUDA	Release 5.0
CULA-Dense	R16a

Table 3: System specifications.

people and bridgeness analyzes the extent to which a given vertex is shared among different communities (Nepusz et al., 2008). Formally, the bridgeness of a vertex i is defined as

$$b_i := 1 - \sqrt{\frac{\widehat{k}}{\widehat{k}-1} \sum_{j=1}^{\widehat{k}} \left(\widehat{\Pi}_i(j) - \frac{1}{\widehat{k}} \right)^2}. \tag{19}$$

Note that centrality measures should be used in conjunction with bridge score to distinguish outliers from genuine bridge nodes (Nepusz et al., 2008). The *degree-corrected bridgeness* is used to evaluate our results and is defined as

$$\mathcal{B}_i := D_i b_i, \tag{20}$$

where D_i is degree of node i.

6. Experimental Results

The specifications of the machine on which we run our code are given in Table 3.

Results on Synthetic Datasets:

We perform experiments for both the stochastic block model ($\alpha_0 = 0$) and the mixed membership model. For the mixed membership model, we set the concentration parameter $\alpha_0 = 1$. We note that the error is around $8\% - 14\%$ and the running times are under a minute, when $n \leq 10000$ and $n \gg k$[5].

We observe that more samples result in a more accurate recovery of memberships which matches intuition and theory. Overall, our learning algorithm performs better in the stochastic block model case than in the mixed membership model case although we note that the accuracy is quite high for practical purposes. Theoretically, this is expected since smaller concentration parameter α_0 is easier for our algorithm to learn (Anandkumar et al., 2013b). Also, our algorithm is scalable to an order of magnitude more in n as illustrated by experiments on real-world large-scale datasets.

5. The code is available at `http://github.com/FurongHuang/Fast-Detection-of-Overlapping-Communities-via-Online-Tensor-Methods`

Note that we threshold the estimated memberships to clean the results. There is a tradeoff between match ratio and average error via different thresholds. In synthetic experiments, the tradeoff is not evident since a perfect matching is always present. However, we need to carefully handle this in experiments involving real data.

Results on Topic Modeling:

We perform experiments for the bag of words data set (Bache and Lichman, 2013) for The New York Times. We set the concentration parameter to be $\alpha_0 = 1$ and observe top recovered words in numerous topics. The results are in Table 4. Many of the results are expected. For example, the top words in topic # 11 are all related to some bad personality.

We also present the words with most spread membership, i.e., words that belong to many topics as in Table 5. As expected, we see minutes, consumer, human, member and so on. These words can appear in a lot of topics, and we expect them to connect topics.

Topic #		Top Words			
1	prompting	complicated	eviscerated	predetermined	lap
	renegotiating	loose	entity	legalese	justice
2	hamstrung	airbrushed	quasi	outsold	fargo
	ennobled	tantalize	irrelevance	noncontroversial	untalented
3	scariest	pest	knowingly	causing	flub
	mesmerize	dawned	millennium	ecological	ecologist
4	reelection	quixotic	arthroscopic	versatility	commanded
	hyperextended	anus	precipitating	underhand	knee
5	believe	signing	ballcarrier	parallel	anomalies
	munching	prorated	unsettle	linebacking	bonus
6	gainfully	settles	narrator	considerable	articles
	narrative	rosier	deviating	protagonist	deductible
7	faithful	betcha	corrupted	inept	retrench
	martialed	winston	dowdy	islamic	corrupting
8	capable	misdeed	dashboard	navigation	opportunistically
	aerodynamic	airbag	system	braking	mph
9	apostles	oracles	believer	deliberately	loafer
	gospel	apt	mobbed	manipulate	dialogue
10	physique	jumping	visualizing	hedgehog	zeitgeist
	belonged	loo	mauling	postproduction	plunk
11	smirky	silly	bad	natured	frat
	thoughtful	freaked	moron	obtuse	stink
12	offsetting	preparing	acknowledgment	agree	misstating
	litigator	prevented	revoked	preseason	entomology
13	undertaken	wilsonian	idealism	brethren	writeoff
	multipolar	hegemonist	multilateral	enlargement	mutating
14	athletically	fictitious	myer	majorleaguebaseball	familiarizing
	resurrect	slug	backslide	superseding	artistically
15	dialog	files	diabolical	lion	town
	password	list	swiss	coldblooded	outgained
16	recessed	phased	butyl	lowlight	balmy
	redlining	prescription	marched	mischaracterization	tertiary
17	sponsor	televise	sponsorship	festival	sullied
	ratification	insinuating	warhead	staged	reconstruct
18	trespasses	buckle	divestment	schoolchild	refuel
	ineffectiveness	coexisted	repentance	divvying	overexposed

Table 4: Top recovered topic groups from the New York Times dataset along with the words present in them.

Keywords
minutes, consumer, human, member, friend, program, board, cell, insurance, shot

Table 5: The top ten words which occur in multiple contexts in the New York Times dataset.

Results on Real-world Graph Datasets: We describe the results on real datasets summarized in Table 6 in detail below. The simulations are summarized in Table 7.

Statistics	Facebook	Yelp	DBLP sub	DBLP
$\|E\|$	766,800	672,515	5,066,510	16,221,000
$\|V\|$	18,163	10,010+28,588	116,317	1,054,066
GD	0.004649	0.000903	0.000749	0.000029
k	360	159	250	6,003
AB	0.5379	0.4281	0.3779	0.2066
ADCB	47.01	30.75	48.41	6.36

Table 6: Summary of real datasets used in our paper: $|V|$ is the number of nodes in the graph, $|E|$ is the number of edges, GD is the graph density given by $\frac{2|E|}{|V|(|V|-1)}$, k is the number of communities, AB is the average bridgeness and ADCB is the average degree-corrected bridgeness(explained in Section 5).

The results are presented in Table 7. We note that our method, in both dense and sparse implementations, performs very well compared to the state-of-the-art variational method. For the Yelp dataset, we have a bipartite graph where the business nodes are on one side and user nodes on the other and use the review stars as the edge weights. In this bipartite setting, the variational code provided by Gopalan et al (Gopalan et al., 2012) does not work on since it is not applicable to non-homophilic models. Our approach does not have this restriction. Note that we use our dense implementation on the GPU to run experiments with large number of communities k as the device implementation is much faster in terms of running time of the STGD step.On the other hand, the sparse implementation on CPU is fast and memory efficient in the case of sparse graphs with a small number of communities while the dense implementation on GPU is faster for denser graphs such as Facebook. Note that data reading time for DBLP is around 4700 seconds, which is not negligible as compared to other datasets (usually within a few seconds). Effectively, our algorithm, excluding the file I/O time, executes within two minutes for $k = 10$ and within ten minutes for $k = 100$.

Interpretation on Yelp Dataset: The ground truth on business attributes such as location and type of business are available (but not provided to our algorithm) and we provide the distribution in Figure 6 on the left side. There is also a natural trade-off between recovery ratio and average error or between attempting to recover all the business communities and the accuracy of recovery. We can either recover top significant communities with high accuracy or recover more with lower accuracy. We demonstrate the trade-off in Figure 6 on the right side.

We select the top ten categories recovered with the lowest error and report the business with highest weights in $\widehat{\Pi}$. Among the matched communities, we find the business with

Data	Method	$\widehat{k}$	Thre	$\mathcal{E}$	$\mathcal{R}(\%)$	Time(s)
FB	Ten(sparse)	10	0.10	0.063	13	35
	Ten(sparse)	100	0.08	0.024	62	309
	Ten(sparse)	100	0.05	0.118	95	309
	Ten(dense)	100	0.100	0.012	39	190
	Ten(dense)	100	0.070	0.019	100	190
	Variational	100	–	0.070	100	10,795
	Ten(dense)	500	0.020	0.014	71	468
	Ten(dense)	500	0.015	0.018	100	468
	Variational	500	–	0.031	100	86,808
YP	Ten(sparse)	10	0.10	0.271	43	10
	Ten(sparse)	100	0.08	0.046	86	287
	Ten(dense)	100	0.100	0.023	43	1,127
	Ten(dense)	100	0.090	0.061	80	1,127
	Ten(dense)	500	0.020	0.064	72	1,706
	Ten(dense)	500	0.015	0.336	100	1,706
DB sub	Ten(dense)	100	0.15	0.072	36	7,664
	Ten(dense)	100	0.09	0.260	80	7,664
	Variational	100	–	7.453	99	69,156
	Ten(dense)	500	0.10	0.010	19	10,157
	Ten(dense)	500	0.04	0.139	89	10,157
	Variational	500	–	16.38	99	558,723
DB	Ten(sparse)	10	0.30	0.103	73	4716
	Ten(sparse)	100	0.08	0.003	57	5407
	Ten(sparse)	100	0.05	0.105	95	5407

Table 7: Yelp, Facebook and DBLP main quantitative evaluation of the tensor method versus the variational method: $\widehat{k}$ is the community number specified to our algorithm, Thre is the threshold for picking significant estimated membership entries. Refer to Table 6 for statistics of the datasets.

Business	**RC**	**Categories**
Four Peaks Brewing Co	735	Restaurants, Bars, American (New), Nightlife, Food, Pubs, Tempe
Pizzeria Bianco	803	Restaurants, Pizza,Phoenix
FEZ	652	Restaurants, Bars, American (New), Nightlife, Mediterranean, Lounges Phoenix
Matt's Big Breakfast	689	Restaurants, Phoenix, Breakfast& Brunch
Cornish Pasty Company	580	Restaurants, Bars, Nightlife, Pubs, Tempe
Postino Arcadia	575	Restaurants, Italian, Wine Bars, Bars, Nightlife, Phoenix
Cibo	594	Restaurants, Italian, Pizza, Sandwiches, Phoenix
Phoenix Airport	862	Hotels & Travel, Phoenix
Gallo Blanco Cafe	549	Restaurants, Mexican, Phoenix
The Parlor	489	Restaurants, Italian, Pizza, Phoenix

Table 8: Top 10 bridging businesses in Yelp and categories they belong to. "RC" denotes review counts for that particular business.

the highest membership weight (Table 9). We can see that most of the "top" recovered businesses are rated high. Many of the categories in the top ten list are restaurants as they have a large number of reviewers. Our method can recover restaurant category with high accuracy, and the specific restaurant in the category is a popular result (with high number of stars). Also, our method can also recover many of the categories with low review counts accurately like hobby shops, yoga, churches, galleries and religious organizations which are

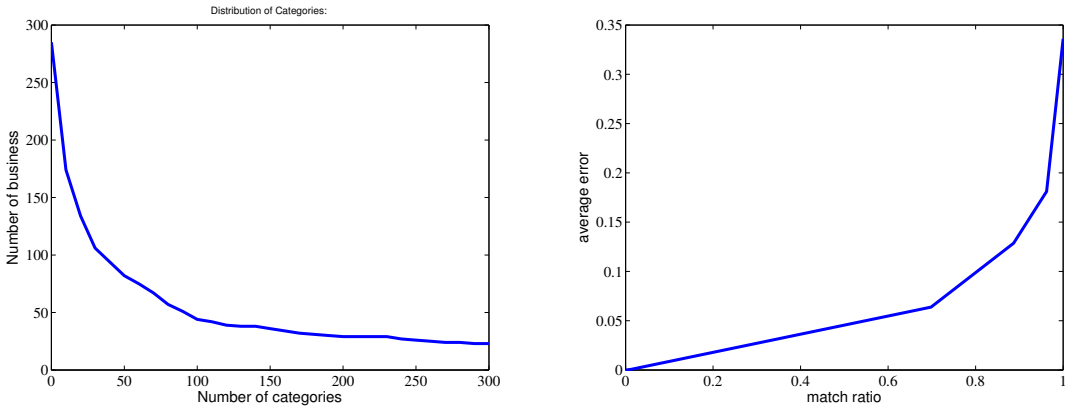


Figure 6: Distribution of business categories (left) and result tradeoff between recovery ratio and error for yelp (right).

the "niche" categories with a dedicated set of reviewers, who mostly do not review other categories.

Category	Business	Star(B)	Star(C)	RC(B)	RC(C)
Latin American	Salvadoreno	4.0	3.94	36	93.8
Gluten Free	P.F. Chang's	3.5	3.72	55	50.6
Hobby Shops	Make Meaning	4.5	4.13	14	7.6
Mass Media	KJZZ 91.5FM	4.0	3.63	13	5.6
Yoga	Sutra Midtown	4.5	4.55	31	12.6
Churches	St Andrew Church	4.5	4.52	3	4.2
Art Galleries	Sette Lisa	4.5	4.48	4	6.6
Libraries	Cholla Branch	4.0	4.00	5	11.2
Religious	St Andrew Church	4.5	4.40	3	4.2
Wickenburg	Taste of Caribbean	4.0	3.66	60	6.7

Table 9: Most accurately recovered categories and businesses with highest membership weights for the Yelp dataset. "Star(B)" denotes the review stars that the business receive and "Star(C)", the average review stars that businesses in that category receive. "RC(B)" denotes the review counts for that business and "RC(C)" , the average review counts in that category.

The top bridging nodes recovered by our method for the Yelp dataset are given in the Table 8. The bridging nodes have multiple attributes typically, the type of business and its location. In addition, the categories may also be hierarchical: within restaurants, different cuisines such as Italian, American or Pizza are recovered by our method. Moreover, restaurants which also function as bars or lounges are also recovered as top bridging nodes in our method. Thus, our method can recover multiple attributes for the businesses efficiently.

Among all 11537 businesses, there are 89.39% of them are still open. We only select those businesses which are still open. There are 285 categories in total. After we remove all the categories having no more than 20 businesses within it, there are 134 categories that remain. We generate community membership matrix for business categories $\Pi_c \in \mathbb{R}^{k_c \times n}$

where $k_c := 134$ is the number of remaining categories and $n := 10141$ is the number of business remaining after removing all the negligible categories. All the businesses collected in the Yelp data are in AZ except 3 of them (one is in CA, one in CO and the other in SC). We remove the three businesses outside AZ. We notice that most of the businesses are spread out in 25 cities. Community membership matrix for location is defined as $\Pi \in \mathbb{R}^{k_l \times n}$ where $k_l := 25$ is the number cities and $n := 10010$ is number of businesses. Distribution of locations are in Table 11. The stars a business receives can vary from 1 (the lowest) to 5 (the highest). The higher the score is, the more satisfied the customers are. The average star score is 3.6745. The distribution is given in Table 10. There are also review counts for each business which are the number of reviews that business receives from all the users. The minimum review counts is 3 and the maximum is 862. The mean of review counts is 20.1929. The preprocessing helps us to pick out top communities.

There are 5 attributes associated with all the 11537 businesses, which are "open", "Categories", "Location", "Review Counts" and "Stars". We model ground truth communities as a combination of "Categories" and "Location". We select business categories with more than 20 members and remove all businesses which are closed. 10010 businesses are remained. Only 28588 users are involved in reviews towards the 10010 businesses. There are 3 attributes associated with all the 28588 users, which are "Female", "Male", "Review Counts" and "Stars". Although we do not directly know the gender information from the dataset, a name-gender guesser[6] is used to estimate gender information using names.

Star Score	Num of businesses	Percentage
1.0	108	0.94%
1.5	170	1.47%
2.0	403	3.49%
2.5	1011	8, 76%
3.0	1511	13.10%
3.5	2639	22.87%
4.0	2674	23.18%
4.5	1748	15.15%
5.0	1273	11.03%

Table 10: Table for distribution of business star scores.

We provide some sample visualization results in Figure 7 for both the ground truth and the estimates from our algorithm. We sub-sample the users and businesses, group the users into male and female categories, and consider nail salon and tire businesses. Analysis of ground truth reveals that nail salon and tire businesses are very discriminative of the user genders, and thus we employ them for visualization. We note that both the nail salon and tire businesses are categorized with high accuracy, while users are categorized with poorer accuracy.

Our algorithm can also recover the attributes of users. However, the ground truth available about users is far more limited than businesses, and we only have information on gender, average review counts and average stars (we infer the gender of the users through

6. `https://github.com/amacinho/Name-Gender-Guesser` by Amac Herdagdelen.

City	State	Num of business
Anthem	AZ	34
Apache Junction	AZ	46
Avondale	AZ	129
Buckeye	AZ	31
Casa Grande	AZ	48
Cave Creek	AZ	65
Chandler	AZ	865
El Mirage	AZ	11
Fountain Hills	AZ	49
Gilbert	AZ	439
Glendale	AZ	611
Goodyear	AZ	126
Laveen	AZ	22
Maricopa	AZ	31
Mesa	AZ	898
Paradise Valley	AZ	57
Peoria	AZ	267
Phoenix	AZ	4155
Queen Creek	AZ	78
Scottsdale	AZ	2026
Sun City	AZ	37
Surprise	AZ	161
Tempe	AZ	1153
Tolleson	AZ	22
Wickenburg	AZ	28

Table 11: Distribution of business locations. Only top cities with more than 10 businesses are presented.

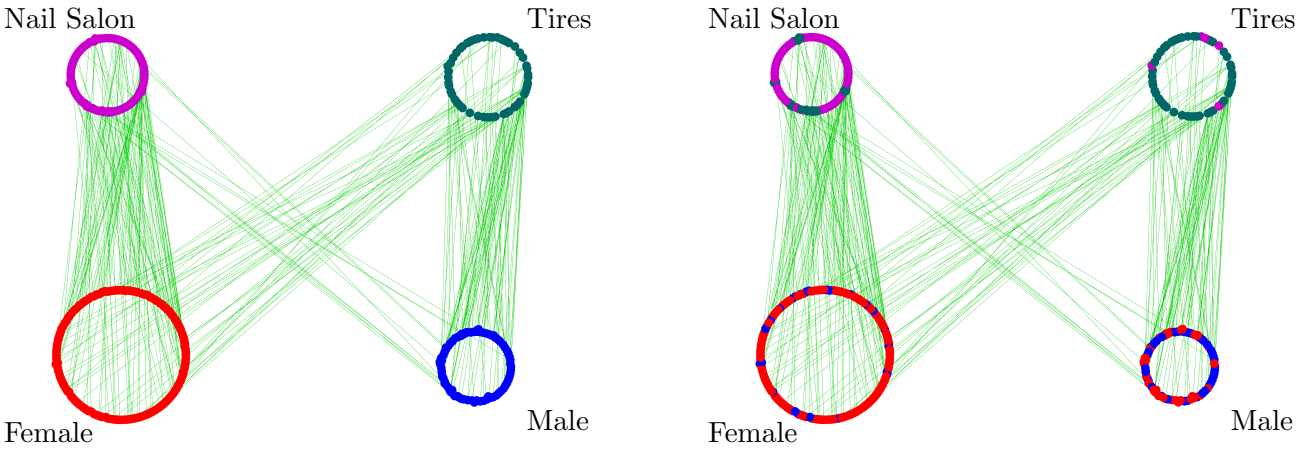


Figure 7: Ground truth (left) vs estimated business and user categories (right). The error in the estimated graph due to misclassification is shown by the mixed colours.

their names). Our algorithm can recover all these attributes. We observe that gender is the hardest to recover while review counts is the easiest. We see that the other user attributes recovered by our algorithm correspond to valuable user information such as their interests, location, age, lifestyle, etc. This is useful, for instance, for businesses studying

the characteristics of their users, for delivering better personalized advertisements for users, and so on.

Facebook Dataset: A snapshot of the Facebook network of UNC (Traud et al., 2010) is provided with user attributes. The ground truth communities are based on user attributes given in the dataset which are not exposed to the algorithm. There are 360 top communities with sufficient (at least 20) users. Our algorithm can recover these attributes with high accuracy; see main paper for our method's results compared with variational inference result (Gopalan et al., 2012).

We also obtain results for a range of values of α_0 (Figure 8). We observe that the recovery ratio improves with larger α_0 since a larger α_0 can recover overlapping communities more efficiently while the error score remains relatively the same.

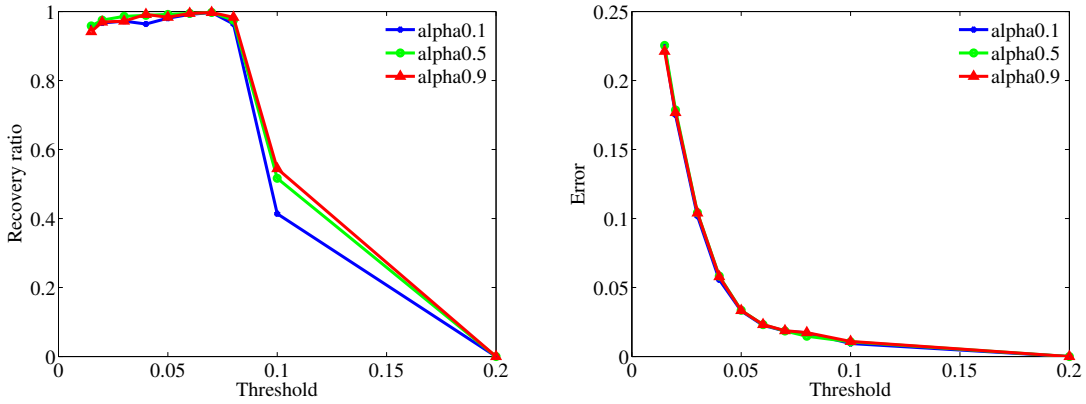


Figure 8: Performance analysis of Facebook dataset under different settings of the concentration parameter (α_0) for $\hat{k} = 100$.

For the Facebook dataset, the top ten communities recovered with lowest error consist of certain high schools, second majors and dorms/houses. We observe that high school attributes are easiest to recover and second major and dorm/house are reasonably easy to recover by looking at the friendship relations in Facebook. This is reasonable: college students from the same high school have a high probability of being friends; so do colleges students from the same dorm.

DBLP Dataset:

The DBLP data contains bibliographic records[7] with various publication venues, such as journals and conferences, which we model as communities. We then consider authors who have published at least one paper in a community (publication venue) as a member of it. Co-authorship is thus modeled as link in the graph in which authors are represented as nodes. In this framework, we could recover the top authors in communities and bridging authors.

7. `http://dblp.uni-trier.de/xml/Dblp.xml`

7. Conclusion

In this paper, we presented a fast and unified moment-based framework for learning overlapping communities as well as topics in a corpus. There are several key insights involved. Firstly, our approach follows from a systematic and guaranteed learning procedure in contrast to several heuristic approaches which may not have strong statistical recovery guarantees. Secondly, though using a moment-based formulation may seem computationally expensive at first sight, implementing implicit "tensor" operations leads to significant speed-ups of the algorithm. Thirdly, employing randomized methods for spectral methods is promising in the computational domain, since the running time can then be significantly reduced.

This paper paves the way for several interesting directions for further research. While our current deployment incorporates community detection in a single graph, extensions to multigraphs and hypergraphs are possible in principle. A careful and efficient implementation for such settings will be useful in a number of applications. It is natural to extend the deployment to even larger datasets by having cloud-based systems. The issue of efficient partitioning of data and reducing communication between the machines becomes significant there. Combining our approach with other simple community detection approaches to gain even more speedups can be explored.

Acknowledgement

The first author is supported by NSF BIGDATA IIS-1251267, the second author is supported in part by UCI graduate fellowship and NSF Award CCF-1219234, and the last author is supported in part by Microsoft Faculty Fellowship, NSF Career award CCF-1254106, NSF Award CCF-1219234, and ARO YIP Award W911NF-13-1-0084. The authors acknowledge insightful discussions with Prem Gopalan, David Mimno, David Blei, Qirong Ho, Eric Xing, Carter Butts, Blake Foster, Rui Wang, Sridhar Mahadevan, and the CULA team. Special thanks to Prem Gopalan and David Mimno for providing the variational code and answering all our questions. The authors also thank Daniel Hsu and Sham Kakade for initial discussions regarding the implementation of the tensor method. We also thank Dan Melzer for helping us with the system-related issues.

Appendix A. Stochastic Updates

After obtaining the whitening matrix, we whiten the data $G_{x,A}^\top$, $G_{x,B}^\top$ and $G_{x,C}^\top$ by linear operations to get y_A^t, y_B^t and $y_C^t \in \mathbb{R}^k$:

$$y_A^t := \left\langle G_{x,A}^\top, W \right\rangle,\ y_B^t := \left\langle Z_B G_{x,B}^\top, W \right\rangle,\ y_C^t := \left\langle Z_C G_{x,C}^\top, W \right\rangle.$$

where $x \in X$ and t denotes the index of the online data.

The stochastic gradient descent algorithm is obtained by taking the derivative of the loss function $\frac{\partial L^t(\mathbf{v})}{\partial v_i}$:

$$\begin{aligned}\frac{\partial L^t(\mathbf{v})}{\partial v_i} =& \theta \sum_{j=1}^{k} \langle v_j, v_i \rangle^2 v_j - \frac{(\alpha_0+1)(\alpha_0+2)}{2} \left\langle v_i, y_A^t \right\rangle \left\langle v_i, y_B^t \right\rangle y_C^t - \alpha_0^2 \left\langle \phi_i^t, \bar{y}_A \right\rangle \left\langle \phi_i^t, \bar{y}_B^t \right\rangle \bar{y}_C \\ &+ \frac{\alpha_0(\alpha_0+1)}{2} \left\langle \phi_i^t, y_A^t \right\rangle \left\langle \phi_i^t, y_B^t \right\rangle \bar{y}_C + \frac{\alpha_0(\alpha_0+1)}{2} \left\langle \phi_i^t, y_A^t \right\rangle \left\langle \phi_i^t, \bar{y}_B \right\rangle y_C \\ &+ \frac{\alpha_0(\alpha_0+1)}{2} \left\langle \phi_i^t, \bar{y}_A \right\rangle \left\langle \phi_i^t, y_B^t \right\rangle y_C\end{aligned}$$

for $i \in [k]$, where y_A^t, y_B^t and y_C^t are the online whitened data points as discussed in the whitening step and θ is a constant factor that we can set.

The iterative updating equation for the stochastic gradient update is given by

$$\phi_i^{t+1} \leftarrow \phi_i^t - \beta^t \frac{\partial L^t}{\partial v_i}\bigg|_{\phi_i^t} \tag{21}$$

for $i \in [k]$, where β^t is the learning rate, ϕ_i^t is the last iteration eigenvector and ϕ_i^t is the updated eigenvector. We update eigenvectors through

$$\phi_i^{t+1} \leftarrow \phi_i^t - \theta\beta^t \sum_{j=1}^{k} \left[\left\langle \phi_j^t, \phi_i^t \right\rangle^2 \phi_j^t \right] + \text{shift}[\beta^t \left\langle \phi_i^t, y_A^t \right\rangle \left\langle \phi_i^t, y_B^t \right\rangle y_C^t] \tag{22}$$

Now we shift the updating steps so that they correspond to the centered Dirichlet moment forms, i.e.,

$$\begin{aligned}&\text{shift}[\beta^t \left\langle \phi_i^t, y_A^t \right\rangle \left\langle \phi_i^t, y_B^t \right\rangle y_C^t] := \beta^t \frac{(\alpha_0+1)(\alpha_0+2)}{2} \left\langle \phi_i^t, y_A^t \right\rangle \left\langle \phi_i^t, y_B^t \right\rangle y_C^t \\ &+ \beta^t \alpha_0^2 \left\langle \phi_i^t, \bar{y}_A \right\rangle \left\langle \phi_i^t, \bar{y}_B \right\rangle \bar{y}_C - \beta^t \frac{\alpha_0(\alpha_0+1)}{2} \left\langle \phi_i^t, y_A^t \right\rangle \left\langle \phi_i^t, y_B^t \right\rangle \bar{y}_C \\ &- \beta^t \frac{\alpha_0(\alpha_0+1)}{2} \left\langle \phi_i^t, y_A^t \right\rangle \left\langle \phi_i^t, \bar{y}_B \right\rangle y_C - \beta^t \frac{\alpha_0(\alpha_0+1)}{2} \left\langle \phi_i^t, \bar{y}_A \right\rangle \left\langle \phi_i^t, y_B^t \right\rangle y_C,\end{aligned} \tag{23}$$

where $\bar{y}_A := \mathbb{E}_t[y_A^t]$ and similarly for $\bar{y}_B$ and $\bar{y}_C$.

Appendix B. Proof of correctness of our algorithm:

We now prove the correctness of our algorithm.

First, we compute M_2 as just

$$\mathbb{E}_x \left[\tilde{G}_{x,C}^\top \otimes \tilde{G}_{x,B}^\top | \Pi_A, \Pi_B, \Pi_C \right]$$

where we define

$$\begin{aligned}\tilde{G}_{x,B}^\top &:= \mathbb{E}_x \left[G_{x,A}^\top \otimes G_{x,C}^\top \,\middle|\, \Pi_A, \Pi_C \right] \left(\mathbb{E}_x \left[G_{x,B}^\top \otimes G_{x,C}^\top \,\middle|\, \Pi_B, \Pi_C \right] \right)^\dagger G_{x,B}^\top \\ \tilde{G}_{x,C}^\top &:= \mathbb{E}_x \left[G_{x,A}^\top \otimes G_{x,B}^\top \,\middle|\, \Pi_A, \Pi_B \right] \left(\mathbb{E}_x \left[G_{x,C}^\top \otimes G_{x,B}^\top \,\middle|\, \Pi_B, \Pi_C \right] \right)^\dagger G_{x,C}^\top.\end{aligned}$$

Define F_A is defined as $F_A := \Pi_A^\top P^\top$, we obtain $M_2 = \mathbb{E}\left[G_{x,A}^\top \otimes G_{x,A}^\top\right] = \Pi_A^\top P^\top \left(\mathbb{E}_x[\pi_x \pi_x^\top]\right) P\Pi_A = F_A \left(\mathbb{E}_x[\pi_x \pi_x^\top]\right) F_A^\top$. Note that P is the community connectivity matrix defined as $P \in [0,1]^{k\times k}$. Now that we know M_2, $\mathbb{E}\left[\pi_i^2\right] = \frac{\alpha_i(\alpha_i+1)}{\alpha_0(\alpha_0+1)}$, and $\mathbb{E}\left[\pi_i \pi_j\right] = \frac{\alpha_i \alpha_j}{\alpha_0(\alpha_0+1)} \forall i \neq j$, we can get the centered second order moments $\text{Pairs}^{\text{Com}}$ as

$$\text{Pairs}^{\text{Com}} := F_A \ \text{diag}\left(\left[\frac{\alpha_1\alpha_1+1}{\alpha_0(\alpha_0+1)}, \ldots, \frac{\alpha_k\alpha_k+1}{\alpha_0(\alpha_0+1)}\right]\right) F_A^\top \tag{24}$$

$$= M_2 - \frac{\alpha_0}{\alpha_0+1} F_A \left(\hat{\alpha}\hat{\alpha}^\top - \ \text{diag}\left(\hat{\alpha}\hat{\alpha}^\top\right)\right) F_A^\top \tag{25}$$

$$= \frac{1}{n_X} \sum_{x\in X} Z_C G_{x,C}^\top G_{x,B} Z_B^\top - \frac{\alpha_0}{\alpha_0+1}\left(\mu_A \mu_A^\top - \ \text{diag}\left(\mu_A \mu_{X\to A}^\top\right)\right) \tag{26}$$

Thus, our whitening matrix is computed. Now, our whitened tensor is $\mathcal{T}$ is given by

$$\mathcal{T} = \mathcal{T}^{\text{Com}}(W,W,W) = \frac{1}{n_X} \sum_x \left[(W^\top F_A \pi_x^{\alpha_0}) \otimes (W^\top F_A \pi_x^{\alpha_0}) \otimes (W^\top F_A \pi_x^{\alpha_0})\right],$$

where $\pi_x^{\alpha_0}$ is the centered vector so that $\mathbb{E}[\pi_x^{\alpha_0} \otimes \pi_x^{\alpha_0} \otimes \pi_x^{\alpha_0}]$ is diagonal. We then apply the stochastic gradient descent technique to decompose the third order moment.

Appendix C. GPU Architecture

The algorithm we propose is very amenable to parallelization and is scalable which makes it suitable to implement on processors with multiple cores in it. Our method consists of simple linear algebraic operations, thus enabling us to utilize *Basic Linear Algebra Subprograms* (BLAS) routines such as BLAS I (vector operations), BLAS II (matrix-vector operations), BLAS III (matrix-matrix operations), Singular Value Decomposition (SVD), and iterative operations such as stochastic gradient descent for tensor decomposition that can easily take advantage of Single Instruction Multiple Data (SIMD) hardware units present in the GPUs. As such, our method is amenable to parallelization and is ideal for GPU-based implementation.

Overview of code design: From a higher level point of view, a typical GPU based computation is a three step process involving data transfer from CPU memory to GPU global memory, operations on the data now present in GPU memory and finally, the result transfer from the GPU memory back to the CPU memory. We use the CULA library for implementing the linear algebraic operations.

GPU compute architecture: The GPUs achieve massive parallelism by having hundreds of homogeneous processing cores integrated on-chip. Massive replication of these cores provides the parallelism needed by the applications that run on the GPUs. These cores, for the Nvidia GPUs, are known as *CUDA cores*, where each core has fully pipelined floating-point and integer arithmetic logic units. In Nvidia's Kepler architecture based GPUs, these CUDA cores are bunched together to form a *Streaming Multiprocessor* (SMX). These SMX units act as the basic building block for Nvidia Kepler GPUs. Each GPU contains multiple SMX units where each SMX unit has 192 single-precision CUDA cores, 64 double-precision

units, 32 special function units, and 32 load/store units for data movement between cores and memory.

Each SMX has L1, shared memory and a read-only data cache that are common to all the CUDA cores in that SMX unit. Moreover, the programmer can choose between different configurations of the shared memory and L1 cache. Kepler GPUs also have an L2 cache memory of about 1.5MB that is common to all the on-chip SMXs. Apart from the above mentioned memories, Kepler based GPU cards come with a large DRAM memory, also known as the global memory, whose size is usually in gigabytes. This global memory is also visible to all the cores. The GPU cards usually do not exist as standalone devices. Rather they are part of a CPU based system, where the CPU and GPU interact with each other via PCI (or PCI Express) bus.

In order to program these massively parallel GPUs, Nvidia provides a framework known as *CUDA* that enables the developers to write programs in languages like C, C++, and Fortran etc. A CUDA program constitutes of functions called *CUDA kernels* that execute across many parallel software threads, where each thread runs on a CUDA core. Thus the GPU's performance and scalability is exploited by the simple partitioning of the algorithm into fixed sized blocks of parallel threads that run on hundreds of CUDA cores. The threads running on an SMX can synchronize and cooperate with each other via the shared memory of that SMX unit and can access the Global memory. Note that the CUDA kernels are launched by the CPU but they get executed on the GPU. Thus compute architecture of the GPU requires CPU to initiate the CUDA kernels.

CUDA enables the programming of Nvidia GPUs by exposing low level API. Apart from CUDA framework, Nvidia provides a wide variety of other tools and also supports third party libraries that can be used to program Nvidia GPUs. Since a major chunk of the scientific computing algorithms is linear algebra based, it is not surprising that the standard linear algebraic solver libraries like BLAS and *Linear Algebra PACKage* (LAPACK) also have their equivalents for Nvidia GPUs in one form or another. Unlike CUDA APIs, such libraries expose APIs at a much higher-level and mask the architectural details of the underlying GPU hardware to some extent thus enabling relatively faster development time.

Considering the tradeoffs between the algorithm's computational requirements, design flexibility, execution speed and development time, we choose *CULA-Dense* as our main implementation library. CULA-Dense provides GPU based implementations of the LAPACK and BLAS libraries for dense linear algebra and contains routines for systems solvers, singular value decompositions, and eigen-problems. Along with the rich set of functions that it offers, CULA provides the flexibility needed by the programmer to rapidly implement the algorithm while maintaining the performance. It hides most of the GPU architecture dependent programming details thus making it possible for rapid prototyping of GPU intensive routines.

The data transfers between the CPU memory and the GPU memory are usually explicitly initiated by CPU and are carried out via the PCI (or PCI Express) bus interconnecting the CPU and the GPU. The movement of data buffers between CPU and GPU is the most taxing in terms of time. The buffer transaction time is shown in the plot in Figure 9. Newer GPUs, like Kepler based GPUs, also support useful features like GPU-GPU direct data transfers without CPU intervention. Our system and software specifications are given in Table 3.

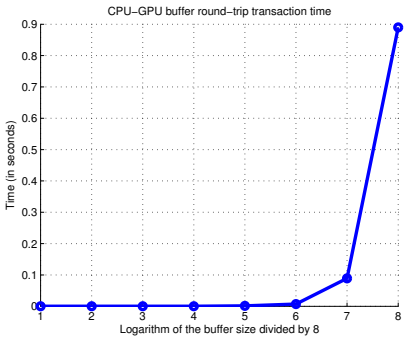


Figure 9: Experimentally measured time taken for buffer transfer between the CPU and the GPU memory in our system.

CULA exposes two important interfaces for GPU programming namely, *standard* and *device*. Using the standard interface, the developer can program without worrying about the underlying architectural details of the GPU as the standard interface takes care of all the data movements, memory allocations in the GPU and synchronization issues. This however comes at a cost. For every standard interface function call the data is moved in and out of the GPU even if the output result of one operation is directly required by the subsequent operation. This unnecessary movement of intermediate data can dramatically impact the performance of the program. In order to avoid this, CULA provides the device interface. We use the device interface for STGD in which the programmer is responsible for data buffer allocations in the GPU memory, the required data movements between the CPU and GPU, and operates only on the data in the GPU. Thus the subroutines of the program that are iterative in nature are good candidates for device implementation.

Pre-processing and post-processing: The pre-processing involves matrices whose leading dimension is of the order of number of nodes. These are implemented using the CULA standard interface BLAS II and BLAS III routines.

Pre-processing requires SVD computations for the Moore-Penrose pseudoinverse calculations. We use CULA SVD routines since these SVD operations are carried out on matrices of moderate size. We further replaced the CULA SVD routines with more scalable SVD and pseudo inverse routines using random projections (Gittens and Mahoney, 2013) to handle larger datasets such as DBLP dataset in our experiment.

After STGD, the community membership matrix estimates are obtained using BLAS III routines provided by the CULA standard interface. The matrices are then used for hypothesis testing to evaluate the algorithm against the ground truth.

Appendix D. Results on Synthetic Datasets

Homophily is an important factor in social interactions (McPherson et al., 2001); the term *homophily* refers to the tendency that actors in the same community interact more than across different communities. Therefore, we assume diagonal dominated community connectivity matrix P with diagonal elements equal to 0.9 and off-diagonal elements equal to 0.1. Note that P need neither be stochastic nor symmetric. Our algorithm allows for randomly

n	k	α_0	Error	Time (secs)
1e2	10	0	0.1200	0.5
1e3	10	0	0.1010	1.2
1e4	10	0	0.0841	43.2
1e2	10	1	0.1455	0.5
1e3	10	1	0.1452	1.2
1e4	10	1	0.1259	42.2

Table 12: Synthetic simulation results for different configurations. Running time is the time taken to run to convergence.

generated community connectivity matrix P with support $[0, 1]$. In this way, we look at general directed social ties among communities.

We perform experiments for both the stochastic block model ($\alpha_0 = 0$) and the mixed membership model. For the mixed membership model, we set the concentration parameter $\alpha_0 = 1$. We note that the error is around $8\% - 14\%$ and the running times are under a minute, when $n \leq 10000$ and $n \gg k$.

The results are given in Table 12. We observe that more samples result in a more accurate recovery of memberships which matches intuition and theory. Overall, our learning algorithm performs better in the stochastic block model case than in the mixed membership model case although we note that the accuracy is quite high for practical purposes. Theoretically, this is expected since smaller concentration parameter α_0 is easier for our algorithm to learn (Anandkumar et al., 2013b). Also, our algorithm is scalable to an order of magnitude more in n as illustrated by experiments on real-world large-scale datasets.

Appendix E. Comparison of Error Scores

Normalized Mutual Information (NMI) score (Lancichinetti et al., 2009) is another popular score which is defined differently for overlapping and non-overlapping community models. For non-overlapping block model, ground truth membership for node i is a discrete k-state categorical variable $\Pi_{\text{block}} \in [k]$ and the estimated membership is a discrete $\widehat{k}$-state categorical variable $\widehat{\Pi}_{\text{block}} \in [\widehat{k}]$. The empirical distribution of ground truth membership categorical variable Π_{block} is easy to obtain. Similarly is the empirical distribution of the estimated membership categorical variable $\widehat{\Pi}_{\text{block}}$. NMI for block model is defined as

$$N_{\text{block}}(\widehat{\Pi}_{\text{block}} : \Pi_{\text{block}}) := \frac{H(\Pi_{\text{block}}) + H(\widehat{\Pi}_{\text{block}}) - H(\Pi_{\text{block}}, \widehat{\Pi}_{\text{block}})}{\Big(H(\Pi_{\text{block}}) + H(\widehat{\Pi}_{\text{block}})\Big)/2}.$$

The NMI for overlapping communities is a binary vector instead of a categorical variable (Lancichinetti et al., 2009). The ground truth membership for node i is a binary vector of length k, $\mathbf{\Pi}_{\text{mix}}$, while the estimated membership for node i is a binary vector of length $\widehat{k}$, $\widehat{\mathbf{\Pi}}_{\text{mix}}$. This notion coincides with one column of our membership matrices $\Pi \in \mathbb{R}^{k \times n}$ and $\widehat{\Pi} \in \mathbb{R}^{\widehat{k} \times n}$ except that our membership matrices are stochastic. In other words, we consider

all the nonzero entries of Π as 1's, then each column of our Π is a sample for Π_{mix}. The m-th entry of this binary vector is the realization of a random variable $\Pi_{\text{mix}_m} = (\mathbf{\Pi}_{\text{mix}})_m$, whose probability distribution is

$$P(\Pi_{\text{mix}_m} = 1) = \frac{n_m}{n}, \quad P(\Pi_{\text{mix}_m} = 0) = 1 - \frac{n_m}{n},$$

where n_m is the number of nodes in community m. The same holds for $\widehat{\Pi}_{\text{mix}_m}$. The normalized conditional entropy between $\mathbf{\Pi}_{\text{mix}}$ and $\widehat{\mathbf{\Pi}}_{\text{mix}}$ is defined as

$$H(\widehat{\mathbf{\Pi}}_{\text{mix}}|\mathbf{\Pi}_{\text{mix}})_{\text{norm}} := \frac{1}{k}\sum_{j\in[k]} \min_{i\in[\widehat{k}]} \frac{H\left(\widehat{\Pi}_{\text{mix}_i}|\Pi_{\text{mix}_j}\right)}{H(\Pi_{\text{mix}_j})} \tag{27}$$

where Π_{mix_j} denotes the j^{th} entry of $\mathbf{\Pi}_{\text{mix}}$ and similarly for $\widehat{\Pi}_{\text{mix}_i}$. The NMI for overlapping community is

$$N_{\text{mix}}(\widehat{\mathbf{\Pi}}_{\text{mix}} : \mathbf{\Pi}_{\text{mix}}) := 1 - \frac{1}{2}\left[H(\mathbf{\Pi}_{\text{mix}}|\widehat{\mathbf{\Pi}}_{\text{mix}})_{\text{norm}} + H(\widehat{\mathbf{\Pi}}_{\text{mix}}|\mathbf{\Pi}_{\text{mix}})_{\text{norm}}\right].$$

There are two aspects in evaluating the error. The first aspect is the l_1 norm error. According to Equation (27), the error function used in NMI score is $\frac{H\left(\widehat{\Pi}_{\text{mix}_i}|\Pi_{\text{mix}_j}\right)}{H(\Pi_{\text{mix}_j})}$. NMI is not suitable for evaluating recovery of different sized communities. In the special case of a pair of extremely sparse and dense membership vectors, depicted in Figure 10, $H(\Pi_{\text{mix}_j})$ is the same for both the dense and the sparse vectors since they are flipped versions of each other (0s flipped to 1s and vice versa). However, the smaller sized community (i.e. the sparser community vector), shown in red in Figure 10, is significantly more difficult to recover than the larger sized community shown in blue in Figure 10. Although this example is an extreme scenario that is not seen in practice, it justifies the drawbacks of the NMI. Thus, NMI is not suitable for evaluating recovery of different sized communities. In contrast, our error function employs a normalized l_1 norm error which penalizes more for larger sized communities than smaller ones.

The second aspect is the error induced by false pairings of estimated and ground-truth communities. NMI score selects only the closest estimated community through normalized conditional entropy minimization and it does not account for statistically significant dependence between an estimated community and multiple ground truth communities and vice-versa, and therefore it underestimates error. However, our error score does not limit to a matching between the estimated and ground truth communities: if an estimated community is found to have statistically significant correlation with multiple ground truth communities (as evaluated by the p-value), we penalize for the error over all such ground truth communities. Thus, our error score is a harsher measure of evaluation than NMI. This notion of "soft-matching" between ground-truth and estimated communities also enables validation of recovery of a combinatorial union of communities instead of single ones.

A number of other scores such as "separability", "density", "cohesiveness" and "clustering coefficient" (Yang and Leskovec, 2012) are non-statistical measures of faithful community recovery. The scores of (Yang and Leskovec, 2012) intrinsically aim to evaluate the

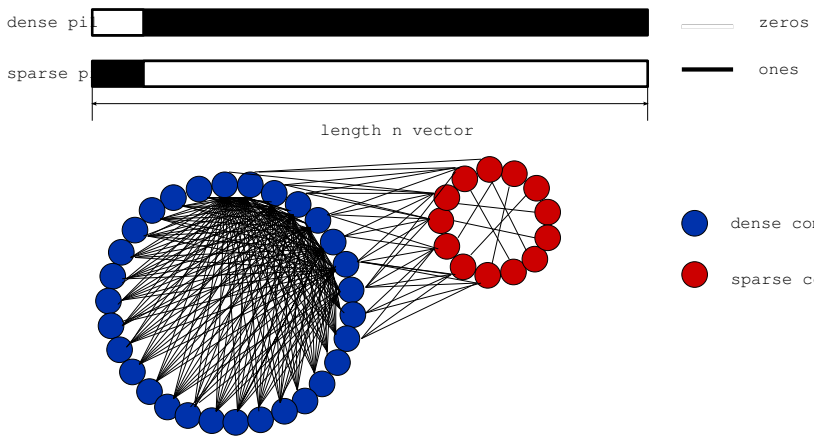


Figure 10: A special case of a pair of extremely dense and sparse communities. Theoretically, the sparse community is more difficult to recover than the dense one. However, the NMI score penalizes both of them equally. Note that for dense Π_1, $P(\Pi_{\text{mix}_1} = 0) = \frac{\#\text{ of 0s in } \Pi_1}{n}$ which is equal to $P(\Pi_{\text{mix}_2} = 1) = \frac{\#\text{ of 1s in } \Pi_2}{n}$. Similarly, $P(\Pi_{\text{mix}_1} = 1) = \frac{\#\text{ of 1s in } \Pi_1}{n}$ which is equal to $P(\Pi_{\text{mix}_2} = 0) = \frac{\#\text{ of 0s in } \Pi_2}{n}$. Therefore, $H(\Pi_{\text{mix}_1}) = H(\Pi_{\text{mix}_2})$.

level of clustering within a community. However our goal is to measure the accuracy of recovery of the communities and not how well-clustered the communities are.

Banerjee and Langford (Banerjee and Langford, 2004) proposed an objective evaluation criterion for clustering which use classification performance as the evaluation measure. In contrast, we look at how well the method performs in recovering the hidden communities, and we are not evaluating predictive performance. Therefore, this measure is not used in our evaluation.

Finally, we note that cophenetic correlation is another statistical score used for evaluating clustering methods, but note that it is only valid for hierarchical clustering and it is a measure of how faithfully a dendrogram preserves the pairwise distances between the original unmodeled data points (Sokal and Rohlf, 1962). Hence, it is not employed in this paper.

References

Edoardo M. Airoldi, David M. Blei, Stephen E. Fienberg, and Eric P. Xing. Mixed membership stochastic blockmodels. *Journal of Machine Learning Research*, 9:1981–2014, June 2008.

A. Anandkumar, R. Ge, D. Hsu, S. M. Kakade, and M. Telgarsky. Tensor decompositions for latent variable models, 2012.

A. Anandkumar, R. Ge, D. Hsu, and S. M. Kakade. A Tensor Spectral Approach to Learning Mixed Membership Community Models. *ArXiv 1302.2684*, Feb. 2013a.

A. Anandkumar, R. Ge, D. Hsu, and S. M. Kakade. A Tensor Spectral Approach to Learning Mixed Membership Community Models. In *Conference on Learning Theory (COLT)*, June 2013b.

Raman Arora, Andrew Cotter, Karen Livescu, and Nathan Srebro. Stochastic optimization for pca and pls. In *Communication, Control, and Computing (Allerton), 2012 50th Annual Allerton Conference on*, pages 861–868, 2012. doi: 10.1109/Allerton.2012.6483308.

K. Bache and M. Lichman. UCI machine learning repository, 2013. URL `http://archive.ics.uci.edu/ml`.

Brett W. Bader, Tamara G. Kolda, et al. Matlab tensor toolbox version 2.5. Available online, January 2012. URL `http://www.sandia.gov/~tgkolda/TensorToolbox/`.

Grey Ballard, Tamara Kolda, and Todd Plantenga. Efficiently computing tensor eigenvalues on a gpu. In *Parallel and Distributed Processing Workshops and Phd Forum (IPDPSW), 2011 IEEE International Symposium on*, pages 1340–1348. IEEE, 2011.

Arindam Banerjee and John Langford. An objective evaluation criterion for clustering. In *Proceedings of the tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 515–520. ACM, 2004.

Michael Berry, Theresa Do, Gavin O'Brien, Vijay Krishna, and Sowmini Varadhan. Svdlibc version 1.4. Available online, 2002. URL `http://tedlab.mit.edu/~dr/SVDLIBC/`.

David M Blei. Probabilistic topic models. *Communications of the ACM*, 55(4):77–84, 2012.

Yudong Chen, Sujay Sanghavi, and Huan Xu. Clustering sparse graphs. *arXiv preprint arXiv:1210.3335*, 2012.

Kenneth L. Clarkson and David P. Woodruff. Low rank approximation and regression in input sparsity time. *CoRR*, abs/1207.6365, 2012.

Paul G Constantine and David F Gleich. Tall and skinny qr factorizations in mapreduce architectures. In *Proceedings of the Second International Workshop on MapReduce and its Applications*, pages 43–50. ACM, 2011.

Barbara Fadem. *High-yield behavioral science*. LWW, 2012.

Dan Feldman, Melanie Schmidt, and Christian Sohler. Turning big data into tiny data: Constant-size coresets for k-means, pca and projective clustering. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1434–1453. SIAM, 2013.

Alex Gittens and Michael W Mahoney. Revisiting the nystrom method for improved large-scale machine learning. *arXiv preprint arXiv:1303.1849*, 2013.

Gene H. Golub and Charles F. Van Loan. *Matrix computations. 4th ed.* Baltimore, MD: The Johns Hopkins University Press, 4th ed. edition, 2013. ISBN 978-1-4214-0794-4/hbk; 978-1-4214-0859-0/ebook.

P. Gopalan, D. Mimno, S. Gerrish, M. Freedman, and D. Blei. Scalable inference of overlapping communities. In *Advances in Neural Information Processing Systems 25*, pages 2258–2266, 2012.

Prem K Gopalan and David M Blei. Efficient discovery of overlapping communities in massive networks. *Proceedings of the National Academy of Sciences*, 110(36):14534–14539, 2013.

Joseph JáJá. *An introduction to parallel algorithms*. Addison Wesley Longman Publishing Co., Inc., 1992.

Ravindran Kannan, Santosh S Vempala, and David P Woodruff. Principal component analysis and higher correlations for distributed data. In *Proceedings of The 27th Conference on Learning Theory*, pages 1040–1057, 2014.

Brian Karrer and Mark EJ Newman. Stochastic blockmodels and community structure in networks. *Physical Review E*, 83(1):016107, 2011.

H.J. Kushner and G. Yin. *Stochastic Approximation and Recursive Algorithms and Applications*. Applications of Mathematics Series. Springer, 2003. ISBN 9780387008943. URL `http://books.google.com/books?id=_ObIieuUJGkC`.

Andrea Lancichinetti and Santo Fortunato. Community detection algorithms: a comparative analysis. *Physical review E*, 80(5):056117, 2009.

Andrea Lancichinetti, Santo Fortunato, and János Kertész. Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics*, 11(3): 033015, 2009.

M. McPherson, L. Smith-Lovin, and J.M. Cook. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, pages 415–444, 2001.

F. McSherry. Spectral partitioning of random graphs. In *FOCS*, 2001.

Andriy Mnih and Ruslan Salakhutdinov. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems*, pages 1257–1264, 2007.

Tamás Nepusz, Andrea Petróczi, László Négyessy, and Fülöp Bazsó. Fuzzy communities and the concept of bridgeness in complex networks. *Physical Review E*, 77(1):016107, 2008.

Erkki Oja and Juha Karhunen. On stochastic approximation of the eigenvectors and eigenvalues of the expectation of a random matrix. *Journal of Mathematical Analysis and Applications*, 106(1):69–84, 1985.

Ruslan Salakhutdinov and Andriy Mnih. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *Proceedings of the 25th International Conference on Machine learning*, pages 880–887. ACM, 2008.

Martin D Schatz, Tze Meng Low, Robert A van de Geijn, and Tamara G Kolda. Exploiting symmetry in tensors for high performance. *arXiv preprint arXiv:1301.7744*, 2013.

Robert R Sokal and F James Rohlf. The comparison of dendrograms by objective methods. *Taxon*, 11(2):33–40, 1962.

Jyothish Soman and Ankur Narang. Fast community detection algorithm with gpus and multicore architectures. In *Parallel & Distributed Processing Symposium (IPDPS), 2011 IEEE International*, pages 568–579. IEEE, 2011.

Korbinian Strimmer. fdrtool: a versatile r package for estimating local and tail area-based false discovery rates. *Bioinformatics*, 24(12):1461–1462, 2008.

Amanda L. Traud, Eric D. Kelsic, Peter J. Mucha, and Mason A. Porter. Comparing community structure to characteristics in online collegiate social networks. SIAM Review, in press (arXiv:0809.0960), 2010.

Jaewon Yang and Jure Leskovec. Defining and evaluating network communities based on ground-truth. In *Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics*, page 3. ACM, 2012.

Yu Zhang and Dit-Yan Yeung. Overlapping community detection via bounded nonnegative matrix tri-factorization. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, pages 606–614, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1462-6. doi: 10.1145/2339530.2339629. URL `http://doi.acm.org/10.1145/2339530.2339629`.

Journal of Machine Learning Research 16 (2015) 2837-2851 Submitted 6/14; Revised 1/15; Published 12/15

Optimal Bayesian Estimation in Random Covariate Design with a Rescaled Gaussian Process Prior

Debdeep Pati DEBDEEP@STAT.FSU.EDU
Department of Statistics
Florida State University
Tallahassee, FL 32306-4330, USA

Anirban Bhattacharya ANIRBANB@STAT.TAMU.EDU
Department of Statistics
Texas A& M University
College Station, TX 77843-3143, USA

Guang Cheng CHENGG@PURDUE.EDU
Department of Statistics
Purdue University
West Lafayette, IN 47907-2066, USA

Editor: Zhihua Zhang

Abstract

In Bayesian nonparametric models, Gaussian processes provide a popular prior choice for regression function estimation. Existing literature on the theoretical investigation of the resulting posterior distribution almost exclusively assume a fixed design for covariates. The only random design result we are aware of (van der Vaart and van Zanten, 2011) assumes the assigned Gaussian process to be supported on the smoothness class specified by the true function with probability one. This is a fairly restrictive assumption as it essentially rules out the Gaussian process prior with a squared exponential kernel when modeling rougher functions. In this article, we show that an appropriate rescaling of the above Gaussian process leads to a rate-optimal posterior distribution even when the covariates are independently realized from a known density on a compact set. The proofs are based on deriving sharp concentration inequalities for frequentist kernel estimators; the results might be of independent interest.

Keywords: Bayesian, convergence rate, Gaussian process, nonparametric regression, random design, rate-optimal

1. Introduction

Gaussian processes (Rasmussen, 2004; Seeger, 2004; Rasmussen and Williams, 2006) are widely used in the machine learning community as a principled probabilistic approach to function estimation. A mean-zero Gaussian process is completely specified by its covariance kernel; popular choices include the squared-exponential and Matérn families. Recently, there has been significant interest in frequentist convergence properties of Bayesian posteriors in Gaussian process models. Ghosal and Roy (2006); Choi and Schervish (2007); Tokdar and Ghosh (2007) established posterior consistency in a variety of settings including nonparametric regression, classification and density estimation. Seeger et al. (2008) used an

information criterion to evaluate closeness of the posterior distribution to the truth; see also van der Vaart and van Zanten (2011). A major focus in the recent literature (van der Vaart and van Zanten, 2007, 2008a, 2009, 2011; Bhattacharya et al., 2014; Shang and Cheng, 2014) has been on deriving the posterior convergence rate (Ghosal et al., 2000), which is defined as the minimum possible sequence $\epsilon_n \to 0$ such that for some constant $M > 0$,

$$E_{\theta_0}\Pi(\|\theta - \theta_0\| < M\epsilon_n \mid \mathcal{D}_n) \to 1, \tag{1}$$

where $\mathcal{D}_n$ denotes the data, θ is the parameter of interest with some known transformation $\Psi(\theta)$ assigned a Gaussian process prior, θ_0 is the true data generating parameter and $\|\cdot\|$ is a distance measure relevant to the statistical problem. In the context of nonparametric regression, classification and density estimation, it has been established that the posterior convergence rate based on appropriate Gaussian process priors coincides with the minimax optimal rate $n^{-\alpha/(2\alpha+d)}$ for d-variate α-smooth functions up to a logarithmic factor (van der Vaart and van Zanten, 2007, 2008a), with rate-adaptivity to the unknown smoothness achieved in van der Vaart and van Zanten (2009); Bhattacharya et al. (2014).

In this paper, we focus on a non-parametric regression problem,

$$Y_i = f(X_i) + \epsilon_i, \quad \epsilon_i \sim \mathrm{N}(0, 1), \tag{2}$$

with f assigned a mean-zero Gaussian process prior. The above-mentioned literature on posterior convergence rates under (2) typically assume that the covariates X_i's are fixed by design, in which the empirical L_2 norm $\|f - f_0\|_{2,n} = (1/n \sum_{i=1}^n |f(x_i) - f_0(x_i)|^2)^{1/2}$ is used as a discrepancy measure in (1). The empirical L_2 norm evaluates the discrepancy of the estimated function from the true function only at the observed data-points and is not suitable to assess out-of-sample predictive performance. In this paper, we consider a *random design* setup where the covariates X_i's are drawn independently from a known distribution q, and derive the posterior convergence rates under an integrated $L_1(q)$ metric:

$$\|f - f_0\|_{1,q} = \int |f(x) - f_0(x)| q(x) dx.$$

The above integrated $L_1(q)$ metric is more relevant for studying statistical efficiency in a random design setting. From a technical standpoint, dealing with the integrated metric is challenging since one cannot directly leverage on properties of multivariate Gaussian distributions as in the case of the empirical L_2 norm to construct "test functions"; a key ingredient in Bayesian asymptotics.

In the frequentist literature, existing results (Baraud, 2002; Brown et al., 2002; Birgé, 2004) on the convergence rates (with respect to an integrated metric) in random design regression require an appropriate lower bound on the smoothness of the underlying true function. For example, Brown et al. (2002); Birgé (2004) assumed that the univariate function f_0 belongs to a Lipschitz class with smoothness index $\alpha > 1/2$. Moreover, Birgé (2004) demonstrated the necessity of the $\alpha > 1/2$ condition by establishing a lower bound for the asymptotic risk for $\alpha \leq 1/2$. Similar lower bound condition will be assumed in our main Bayesian Theorem as well.

As far as we are aware, the only Bayesian literature considering the random design setting in (2) is van der Vaart and van Zanten (2011) who assigned Gaussian processes

with Matérn or squared exponential kernels. Specifically, they obtained an optimal rate $n^{-\alpha/(2\alpha+d)}$ (up to a logarithmic factor, with respect to $L_2(q)$ norm) under a particularly strong assumption that the Gaussian process prior assigns probability one to the smoothness class containing the true function. Since the squared-exponential kernel has infinitely smooth sample paths, their result only delivers the optimal rate for analytic functions, but provides a highly suboptimal $(\log n)^{-t}$ rate for α-smooth functions. This significantly limits the applicability of their result in the sense that it rules out the use of a squared-exponential kernel for less smooth (but more commonly used) functions. An influential idea developed in van der Vaart and van Zanten (2007, 2009) is to scale the sample paths of a Gaussian process with a squared-exponential kernel to enable better approximation of α-smooth functions. The scaling is typically dependent on the smoothness of the true function and the sample size. However, Theorem 2 of van der Vaart and van Zanten (2011) is applicable only to priors without scaling. This is not evident from the statement of their theorem, but a closer inspection of their proof (ref. Page 2113) reveals that they have assumed $\tau^2 := \int \|f\|^2_{\alpha|\infty} \, d\Pi(f)$ to be a global constant for every f in the support of the prior. This may not hold for a rescaled Gaussian process.

In this article, we show that an appropriately rescaled Gaussian process prior with a squared-exponential covariance kernel leads to a rate-optimal posterior distribution (with respect to $L_1(q)$ norm) for any α-smooth function d-variate f_0 in a random design setting if $\alpha > d/2$. While van der Vaart and van Zanten (2011) conjectured (see pp. 2103 after Theorem 2) that their smoothness assumption on the prior is unavoidable for the $L_2(q)$ norm, our result shows that this situation turns out to be different under the $L_1(q)$ norm. Specifically, we develop exponentially consistent test functions under the $L_1(q)$ norm using concentration inequalities for the Nadaraya–Watson kernel estimator. Existence of such test functions plays a key role in Bayesian asymptotic theory (Ghosal et al., 2000). For example, the classical Birgé – Le Cam testing theory (Birgé, 1984; Le Cam, 1986) for the Hellinger metric provides appropriate tests in a wide variety of settings. Giné and Nickl (2011) proposed an alternative framework for constructing tests based on concentration inequalities of frequentist estimators which is particularly useful for stronger norms; see also Ray (2013); Pati et al. (2014); Shang and Cheng (2014) for similar ideas in different contexts.

2. Posterior Convergence in Random Design Regression

2.1 Notations

Let $C[0,1]^d$ and $C^\alpha[0,1]^d$ denote the space of continuous functions and the Hölder space of α-smooth functions $f : [0,1]^d \to \mathbb{R}$, respectively, endowed with the supremum norm $\|f\|_\infty = \sup_{t\in[0,1]^d} |f(t)|$. For $\alpha > 0$, the Hölder space $C^\alpha[0,1]^d$ consists of functions $f \in C[0,1]^d$ that have bounded mixed partial derivatives up to order $\lfloor\alpha\rfloor$, with the partial derivatives of order $\lfloor\alpha\rfloor$ being Lipschitz continuous of order $\alpha - \lfloor\alpha\rfloor$. Let $\|\cdot\|_1$ and $\|\cdot\|_2$ respectively denote the L_1 and L_2 norm on $[0,1]^d$ with respect to the Lebesgue measure (i.e., the uniform distribution). To distinguish the L_2 norm with respect to the Lebesgue measure on $\mathbb{R}^d$, we use the notation $\|\cdot\|_{2,d}$.

We write "$\precsim$" for inequality up to a constant multiple. Let $\phi(t) = (2\pi)^{-1/2}\exp(-t^2/2)$ denote the standard normal density, and let $\phi_n(x) = \prod_{i=1}^n \phi(x_i)$ for $x \in \mathbb{R}^n$. Let a star denote a convolution, i.e., $f_1 \star f_2(y) = \int f_1(y-t)f_2(t)dt$. We denote the Fourier transform of f, whenever defined, by $\hat{f}$, with $\hat{f}(\lambda) = (2\pi)^{-d}\int \exp(i\langle\lambda, t\rangle)f(t)dt$, where $\langle\lambda, t\rangle$ denotes the complex inner product. Under this convention, the inverse Fourier transform $f(t) = \int \exp(-i\langle\lambda, t\rangle)\hat{f}(\lambda)d\lambda$ and $\hat{h} = (2\pi)^d\hat{f}\hat{g}$ when $h = f \star g$.

Throughout $C, C', C_1, C_2, \ldots$ are generically used to denote positive constants whose values might change from one line to another, but are independent from everything else. $Z_{1:n}$ is used as a shorthand for $Z_1, \ldots, Z_n$.

In the sequel, we consider a Gaussian process prior Π on the regression function f with $\mathbb{E}f(x) = 0$ and covariance kernel $c(x, x') = \mathrm{cov}(f(x), f(x'))$. In particular, we focus on the squared-exponential kernel $c_a(x, x') = \exp(-a^2\|x - x'\|^2)$ indexed by an "inverse-bandwidth" parameter a. We next recall some important facts relevant to our setting from van der Vaart and van Zanten (2009) regarding the spectral measure and reproducing kernel Hilbert space of Gaussian process priors. For the squared-exponential kernel c_a, the spectral measure μ_a admits a density ω_a with respect to Lebesgue measure, where $\omega_a(\lambda) = a^{-d}\omega(\lambda/a)$, with $\omega(\lambda) = \exp(-\|\lambda\|^2/4)/(2^d\pi^{d/2})$. The reproducing kernel Hilbert space $\mathbb{H}^a$ associated with a Gaussian process prior Π consists of (real parts of) functions $h(t) = \int \exp(i\langle\lambda, t\rangle)\xi(\lambda)d\mu_a(\lambda)$, where μ_a is the spectral measure of Π and $\xi \in L_2(\mu_a)$. The squared Hilbert space norm of h above is given by $\|h\|^2_{\mathbb{H}^a} = \left\|\xi\omega_a^{1/2}\right\|^2_{2,d} = \int \xi^2(\lambda)\omega_a(\lambda)d\lambda$; let $\mathbb{H}^a_1$ denote the unit ball of the reproducing kernel Hilbert space $\{h \in \mathbb{H}^a : \|h\|_{\mathbb{H}^a} \leq 1\}$. Finally, let $\mathbb{B}_1$ denote the unit ball of $C[0,1]^d$ with respect to the supremum norm. For a detailed review of reproducing kernel Hilbert space of Gaussian process priors and connections with posterior contraction rates, kindly refer to van der Vaart and van Zanten (2008b).

2.2 Main Result

Consider the nonparametric regression model (2). We assume a random design setup, where given the regression function $f : [0,1]^d \to \mathbb{R}$, the data $(X_1, Y_1), \ldots, (X_n, Y_n)$ are independently generated, with X_i having a density q on $[0,1]^d$ that is bounded away from zero and infinity. Let $q(y, x) = q(y \mid x)q(x)$ denote the joint density of (Y, X) given f, where $q(y \mid x) = \phi\{y - f(x)\}$. The joint data likelihood given f is therefore

$$q^{(n)}(Y_{1:n}, X_{1:n} \mid f) = \prod_{i=1}^n q(Y_i, X_i) = \prod_{i=1}^n \phi\{Y_i - f(X_i)\}q(X_i).$$

Similarly, we define $q^{(n)}(Y_{1:n} \mid X_{1:n}, f)$ and $q^{(n)}(X_{1:n})$ as the density of $(Y_{1:n} \mid X_{1:n}, f)$ and $X_{1:n}$ respectively. Let $\mathbb{E}^f_{Y,X}(\mathbb{P}^f_{Y,X})$ denote an expectation (probability) with respect to $q^{(n)}(Y_{1:n}, X_{1:n} \mid f)$. Similarly define $\mathbb{E}^f_{Y|X}(\mathbb{P}^f_{Y|X})$ and $\mathbb{E}^f_X(\mathbb{P}^f_X)$. When f is clear from the context, we shall drop it from the superscript.

We assume a mean zero Gaussian process prior Π on f with a squared exponential kernel $\exp(-a_n^2\|x - x'\|^2)$ and denote the corresponding posterior measure by $\Pi(\cdot \mid Y_{1:n}, X_{1:n})$, so that

$$\Pi(f \mid Y_{1:n}, X_{1:n}) \propto q^{(n)}(Y_{1:n} \mid X_{1:n}, f)\,\Pi(f).$$

Assuming the true regression function is f_0, we study concentration of the posterior $\Pi(\cdot \mid Y_{1:n}, X_{1:n})$ in an $L_1(q)$ neighborhood of f_0.

Theorem 1 *Assume that $f_0 \in C^\alpha[0,1]^d$ with $\alpha > d/2$ and Π is a mean-zero Gaussian process prior with a squared exponential covariance kernel $c(x,x') = \exp(-a_n^2 \|x - x'\|^2)$. Set $a_n = n^{1/(2\alpha+d)}$. Then with $\epsilon_n = n^{-\alpha/(2\alpha+d)} \log^{3t_1/2} n$ for $t_1 \geq (d+1)/2$, and some fixed sufficiently large constant $M > 0$,*

$$\mathbb{E}^{f_0}_{Y,X} \Pi\big(\|f - f_0\|_{1,q} > M\epsilon_n \mid Y_{1:n}, X_{1:n} \big) \to 0. \tag{3}$$

As stated previously, the condition $\alpha > d/2$ is necessary to obtain the optimal rate. van der Vaart and van Zanten (2009) showed that the squared-exponential covariance kernel without rescaling leads to a very slow $(\log n)^{-l}$ contraction rate for α-smooth functions both in the fixed and random design settings. This is not surprising as the sample paths of such a GP are analytic. The effect of scaling the prior using the "inverse bandwidth" a to yield the optimal posterior concentration was first noted by van der Vaart and van Zanten (2007) in a fixed design context, who showed (for $d = 1$) that a deterministic scaling $a_n = n^{1/(2\alpha+1)}$ produces priors that are suitable for modeling α-smooth functions. Theorem 1 assures that the same rescaling idea continues to work in the *random* design setting for an integrated L_1 norm.

The optimal rescaling in Theorem 1 requires knowledge of the true smoothness α. If there is a mismatch between the prior regularity and the function smoothness, one would typically expect a sub-optimal rate. Corollary 2 quantifies this fact; while we only derive an upper bound to the posterior convergence rate, such bounds are usually tight (van der Vaart and van Zanten, 2009). In absence of any prior knowledge regarding the smoothness, one may resort to an empirical or fully Bayes approach as in van der Vaart and van Zanten (2009); Szabó et al. (2013). The related theoretical investigation will be considerably harder in such cases.

Corollary 2 *Under the conditions of Theorem 1, if $a_n = n^{1/(2\beta+d)}$ for $\beta > d/2, \beta \neq \alpha$, the conclusion of Theorem 1 holds with $\epsilon_n = n^{-\alpha\wedge\beta/(2\beta+d)} \log^{3t_1/2} n$ for $t_1 \geq d/(4-2\kappa)$ for any $0 < \kappa < 2$.*

Observe that the optimal rate $n^{-\alpha/(2\alpha+d)}$ is attained (upto logarithmic terms) if and only if $\alpha = \beta$. A scaling $n^{1/(2\beta+d)}$ for $\beta < \alpha$ makes the prior rougher compared to the true function while $\beta > \alpha$ renders smoother prior realizations. In both cases, sub-optimal rates are obtained. This is in accordance with the findings for GP priors with Matérn covariance kernel; refer to Theorem 5 in van der Vaart and van Zanten (2011).

Note that by taking κ very close to 0, we can improve on the power of the $\log n$ term in Corollary 2 from that in Theorem 1. The difference in the power of $\log n$ stems from the fact that the corollary only targets sub-optimal rates as opposed to Theorem 1. Hence a portion of the power of $\log n$ can be eliminated in Corollary 2.

2.3 Contributions Beyond Literature

The proof of Theorem 1 follows from a general set of sufficient conditions for posterior concentration in model (2); kindly refer to Theorem 3 stated in the next Section. In

particular, we exploit concentration inequalities for suitable kernel estimators to construct the aforementioned exponentially consistent sequence of test functions. Such techniques have been used previously to show convergence rates in density estimation (Giné and Nickl, 2011) and in linear inverse problems (Ray, 2013). Their techniques do not directly apply to our case partly due to the lack of concentration bounds for kernel based estimators. Giné and Nickl (2011); Ray (2013) construct estimators based on truncated spectral representations which are well suited to sieve priors. However, to deal with a Gaussian process prior with a squared-exponential covariance kernel, we need to construct test functions based on the Nadaraya–Watson kernel estimator and derive sharp concentration bounds for this class of estimators in Lemma 4 and 5.

The choice of the norm dictating the neighborhood around the true parameter plays a critical role in Bayesian asymptotics. A fundamental tool for relating the likelihood ratio with the neighborhood in consideration is a sequence of exponentially consistent test functions (Ghosal et al., 2000). In a regression setting, such test functions are guaranteed to exist for the empirical L_2 norm by exploiting a direct relation between the empirical L_2 norm and the likelihood ratio of the multivariate Gaussian densities involved; refer to Section 4 of van der Vaart and van Zanten (2011). However, the integrated norm involves covariate points different from the observations, which makes the problem more challenging. van der Vaart and van Zanten (2011) applied Bernstein's inequality to extrapolate to the $L_2(q)$ norm from the empirical L_2 norm. However, as stated in the Introduction, their approach only works for priors that are supported on the true smoothness class with probability one.

Among other related work, Section 4 of Kleijn and van der Vaart (2006) considers random design regression, where a correspondence between the Kullback–Leibler and $L_2(q)$ neighborhood is established to derive the test function, assuming the prior support consists of uniformly bounded functions. However, this assumption does not hold for the rescaled Gaussian process prior in Theorem 1. In particular, the *sieves* constructed in van der Vaart and van Zanten (2007) of the form $M_n\mathbb{H}_1^{a_n} + \epsilon_n\mathbb{B}_1$ with $M_n \to \infty$ do not correspond to sup-norm bounded subsets of $C[0,1]^d$.

We comment here that convergence in the integrated metric has been settled in the binary regression setting. Using a logistic link function, a direct agreement can be established between the integrated L_1 metric on the function space and the Hellinger distance between the resulting densities arising from the Bernoulli likelihood; see for example, Section 3.2 of van der Vaart and van Zanten (2008a). Second, in this paper we implicitly refer to Gaussian processes which are specified by a kernel function; specifically, kernel functions which do not admit a finite series representation, such as the squared-exponential kernel. If a Gaussian process is specified via a truncated orthogonal series representation with independent Gaussian priors on the coefficients, the integrated metric can be related to the L_2 norm of the coefficient vector (Bontemps, 2011).

3. Auxiliary Results

We now state a general theorem which presents a set of sufficient conditions for proving Theorem 1. From now onwards, we shall assume the covariate distribution q to be a uniform distribution on $[0,1]^d$ for notational simplicity; modifying our construction to a general q, which is bounded from above and below, is straightforward. The $L_1(q)$ norm $\|\cdot\|_{1,q}$ with q

the uniform distribution on $[0,1]^d$ shall be simply denoted by $\|\cdot\|_1$ following our convention in Section 2.1. A proof of Theorem 3 can be found in the Appendix.

Theorem 3 *Let ϵ'_n, δ_n be sequences such that $\epsilon'_n, \delta_n \to 0$ and $n\epsilon'^2_n, n\delta^2_n \to \infty$. Let $U_n = \{f : \|f - f_0\|_1 > M\epsilon'_n\}$ for some fixed $M > 0$. Suppose that there exists a sequence of estimators $\tilde{f}_n$ for f based on $(Y_{1:n}, X_{1:n})$ and a sequence of subsets/sieves $\mathcal{P}_n$ of $C[0,1]^d$ such that*

$$\Pi(\mathcal{P}_n^c) \leq \exp\{-(C+4)n\delta_n^2\}, \tag{PCS}$$

$$\left\|\mathbb{E}^{f_0}_{Y,X}\tilde{f}_n - f_0\right\|_1 < \epsilon'_n, \tag{BT}$$

$$\mathbb{P}^{f_0}_{Y,X}\left(\left\|\tilde{f}_n - \mathbb{E}^{f_0}_{Y,X}\tilde{f}_n\right\|_1 > \epsilon'_n\right) \leq \exp\{-(C+4)n\delta_n^2\}, \tag{DT}$$

$$\sup_{f\in\mathcal{P}_n\cap U_n}\left\|\mathbb{E}^{f}_{Y,X}\tilde{f}_n - f\right\|_1 < \epsilon'_n, \tag{BS}$$

$$\sup_{f\in\mathcal{P}_n\cap U_n}\mathbb{P}^{f}_{Y,X}\left(\left\|\tilde{f}_n - \mathbb{E}^{f}_{Y,X}\tilde{f}_n\right\|_1 > \epsilon'_n\right) \leq \exp\{-(C+4)n\delta_n^2\}, \tag{DS}$$

$$\Pi\Big(\|f - f_0\|_\infty \leq \delta_n\Big) \geq \exp\{-n\delta_n^2\}. \tag{PCN}$$

Then, $\mathbb{E}^{f_0}_{Y,X}\Pi\big(U_n \mid Y_{1:n}, X_{1:n}\big) \to 0$.

Condition (PCS) implies that the prior probability of the complement of the sieve $\mathcal{P}_n$ is exponentially small. Condition (BT) assumes a sufficiently accurate estimator $\tilde{f}_n$ with bias smaller than ϵ_n at f_0 while (DT) assumes an exponential concentration bound of $\tilde{f}_n$ from its expectation under $q^{(n)}(\cdot \mid f_0)$. (BS) and (DS) assume similar conditions as (BT) and (DT) under $q^{(n)}(\cdot \mid f)$ for any $f \in \mathcal{P}_n \cap U_n$. The conditions (BT), (DT); (BS), (DS) jointly guarantee the existence of exponentially consistent test functions; see Lemma 9 in the Appendix. Condition (PCN) assumes that the prior Π places "enough" mass in an ϵ_n-neighborhood of the truth f_0 in terms of the sup-norm.

3.1 Verifying the Conditions of Theorem 3 to Prove Theorem 1

Letting $\delta_n = \epsilon'_n = \epsilon_n$ with ϵ'_n and ϵ_n as in the statement of Theorem 3 and Theorem 1 respectively, we now proceed to construct $\mathcal{P}_n$ and $\tilde{f}_n$ that satisfy the conditions of Theorem 3. While we choose the same sieve as in van der Vaart and van Zanten (2007), part of the technical challenge lies in the fact that the concentration bounds need to be derived not just for the truth, but rather for every function in the sieve. This requires precise control on the *size* of the functions in the sieve $\mathcal{P}_n$. We show in Proposition 7 below that the functions in the chosen sieve are uniformly bounded in L_2 norm, although they are unbounded in the supremum norm.

Let $\psi : \mathbb{R}^d \to \mathbb{C}$ be a function such that $\int \psi(t)dt = 1$, $\int t^k\psi(t)dt = 0$ for any non-zero multi-index $k = (k_1, \ldots, k_d)$ of non-zero integers, $\int |t|^{\max\{\alpha,2\}}|\psi(t)|dt < \infty$, and the functions $|\hat{\psi}|/\omega$ and $|\hat{\psi}|^2/\omega$ are uniformly bounded; see proof of Lemma 4.3 in van der Vaart and van Zanten (2009). Define,

$$\tilde{f}_n(x) = \frac{1}{n}\sum_{i=1}^{n}\psi_{\sigma_n}(x - X_i)Y_i, \tag{4}$$

where $\psi_\sigma(t) = \sigma^{-d}\psi(t/\sigma)$ for $\sigma > 0$ and set $\sigma_n = n^{-1/(2\alpha+d)}\log^{-t_2} n$, $t_2 = 1/(2-\kappa)$ for some $0 < \kappa < 1$. Next, with $M_n = a_n^{d/2}$, set

$$\mathcal{P}_n = M_n \mathbb{H}_1^{a_n} + \epsilon_n \mathbb{B}_1. \tag{5}$$

Assume $f_0 \in C^\alpha[0,1]^d$. Let $\tilde{f}_n$ and $\mathcal{P}_n$ be as in (4) and (5) respectively. We show below that the conditions of Theorem 3 are satisfied with $\epsilon_n = n^{-\alpha/(2\alpha+d)}\log^{t_1} n$ for $t_1 \geq \max\{t_2 d/2, (d+1)/2\} = (d+1)/2$, provided $\alpha > d/2$. (PCS) follow from the proof of Theorem 3.1 in van der Vaart and van Zanten (2009). To verify (PCN), observe from the proof of Theorem 3.1 in van der Vaart and van Zanten (2009) that with $a_n = n^{1/(2\alpha+d)}$,

$$\Pi(\|f - f_0\|_\infty \leq \delta_n) \geq \exp\{-n^{d/(2\alpha+d)}(\log n)^{d+1}\}$$

for $\delta_n \geq n^{-\alpha/(2\alpha+d)}$. Hence (PCN) is satisfied with $\delta_n = \epsilon_n$.

We verify (BS) and (DS); the verifications of (BT) and (DT) follow along similar lines.

We first show that (DS) holds. Fix $f \in \mathcal{P}_n \cap U_n$. We drop the superscript f from $\mathbb{E}^f$ in the sequel. Let $f_n(x) = \psi_{\sigma_n} \star f(x) = \int \psi_{\sigma_n}(x-t)f(t)dt$ and $f_n^X(x) = n^{-1}\sum_{i=1}^n \psi_{\sigma_n}(x - X_i)f(X_i)$. Observe that $\mathbb{E}_{Y,X}\tilde{f}_n = f_n$ and $\mathbb{E}_{Y|X}\tilde{f}_n = f_n^X$. Then,

$$\begin{aligned}
&\mathbb{P}_{Y,X}\left(\left\|\tilde{f}_n - \mathbb{E}_{Y,X}\tilde{f}_n\right\|_1 > \epsilon_n \right) = \mathbb{P}_{Y,X}\left(\left\|\tilde{f}_n - f_n\right\|_1 > \epsilon_n \right) \\
&\leq \mathbb{P}_{Y,X}\left(\left\|\tilde{f}_n - f_n\right\|_1 > \epsilon_n, \left\|f_n^X - f_n\right\|_1 \leq \epsilon_n/2 \right) + \mathbb{P}_X\left(\left\|f_n^X - f_n\right\|_1 \geq \epsilon_n/2 \right) \\
&\leq \mathbb{E}_X \mathbb{P}_{Y|X}\left(\left\|\tilde{f}_n - f_n^X\right\|_1 \geq \epsilon_n/2 \mid X_{1:n} \right) + \mathbb{P}_X\left(\left\|f_n^X - f_n\right\|_1 \geq \epsilon_n/2 \right).
\end{aligned} \tag{6}$$

Lemmata 4 and 5 below deliver the desired bounds for the two terms appearing in (6).

Lemma 4 *Under conditions of Theorem 1,*

$$\mathbb{P}_{Y|X}\left(\left\|\tilde{f}_n - f_n^X\right\|_1 \geq \epsilon_n/2 \mid X_{1:n} \right) \leq \exp(-Cn\epsilon_n^2) \quad a.s.$$

for some constant $C > 0$.

Proof For simplicity of notation, we suppress the term "a.s." in the displays that follow. Let $T(x) = n(\tilde{f}_n - f_n^X)(x) = \sum_{i=1}^n \psi_{\sigma_n}(x - X_i)Z_i$, where $Z_i = Y_i - f(X_i)$ with $Z_{1:n} \mid X_{1:n}, f \sim \mathrm{N}_n(0, \mathrm{I})$. Given $X_{1:n}$, T is a random element of $L_1[0,1]^d$ and $\|T\|_1$ is a non-negative random variable. By the Hahn–Banach theorem, there exists a bounded linear functional G on $L_\infty[0,1]^d$ such that $G(h) = \int T(x)h(x)dx$ for all $h \in L_\infty[0,1]^d$ and $\|T\|_1 = \|G\|_{\mathcal{F}}$, where $\|G\|_{\mathcal{F}} = \sup_{h\in\mathcal{F}} |G(h)|$ and $\mathcal{F}$ is a countable dense subset of $\{h \in L_\infty[0,1]^d : \|h\|_\infty \leq 1\}$.

By definition, $G(h) = \sum_{i=1}^n a_i Z_i$, where $a_i = \int \psi_{\sigma_n}(x - X_i)h(x)dx$. Thus, given $X_{1:n}$, $\{G(h) : h \in \mathcal{F}\}$ is a Gaussian process and

$$\mathbb{P}_{Y|X}\left(\left\|\tilde{f}_n - f_n^X\right\|_1 \geq \epsilon_n/2 \mid X_{1:n} \right) = \mathbb{P}_{Y|X}\left(\|G\|_{\mathcal{F}} \geq n\epsilon_n/2 \mid X_{1:n} \right). \tag{7}$$

By Borell's inequality (Adler, 1990),

$$\mathbb{P}_{Y|X}\big(\|G\|_{\mathcal{F}} - \mathbb{E}_{Y|X} \|G\|_{\mathcal{F}} \geq t \mid X_{1:n}\big) \leq 2\exp\{-t^2/(2\sigma_{\mathcal{F}}^2)\}, \tag{8}$$

where $\sigma_{\mathcal{F}}^2 = \sup_{h\in\mathcal{F}} \mathbb{E}_{Y|X} G(h)^2$. We now proceed to estimate $\sigma_{\mathcal{F}}^2$ and $\mathbb{E}_{Y|X} \|G\|_{\mathcal{F}}$. For any $h \in \mathcal{F}$,

$$\begin{aligned}\mathbb{E}_{Y|X} G(h)^2 &= \sum_{i=1}^{n} \left\{ \int_{[0,1]^d} \psi_{\sigma_n}(x - X_i) h(x) dx \right\}^2 \\ &\leq \|h\|_\infty^2 \sum_{i=1}^{n} \left\{ \int_{[0,1]^d} |\psi_{\sigma_n}(x - X_i)|\, dx \right\}^2 \leq C_1 n, \end{aligned} \tag{9}$$

where $C_1 = \int |\psi(t)| dt$. Hence, $\sigma_{\mathcal{F}}^2 \leq C_1 n$.

We next bound $\mathbb{E}_{Y|X} \|G\|_{\mathcal{F}} = \mathbb{E}_{Y|X} \|T\|_1$. By Jensen's inequality, $\mathbb{E}_{Y|X} \|T\|_1 \leq (\mathbb{E}_{Y|X} \|T\|_1^2)^{1/2}$. Further,

$$(\mathbb{E}_{Y|X} \|T\|_1^2)^{1/2} = \left[\int \left\{ \int |T(x)| dx \right\}^2 \phi_n(z) dz \right]^{1/2} \leq \int \left\{ \int T(x)^2 \phi_n(z) dz \right\}^{1/2} dx,$$

where the above inequality follows from an integral version of Minkowski's inequality. Recalling $T(x) = \sum_{i=1}^n \psi_{\sigma_n}(x - X_i) Z_i$, $\int T(x)^2 \phi_n(z) dz = \mathbb{E}[T(x)^2 \mid X_{1:n}] = \sum_{i=1}^n \psi_{\sigma_n}(x - X_i)^2$. Substituting this in the above display and using Jensen's inequality one more time, we get

$$\begin{aligned}\mathbb{E}_{Y|X} \|T\|_1 &\leq \int \left\{ \sum_{i=1}^{n} \psi_{\sigma_n}(x - X_i)^2 \right\}^{1/2} dx \\ &\leq \left\{ \sum_{i=1}^{n} \int \psi_{\sigma_n}(x - X_i)^2 dx \right\}^{1/2} \leq C_2 (n/\sigma_n^d)^{1/2}, \end{aligned} \tag{10}$$

where $C_2 = \{\int \psi(t)^2 dt\}^{1/2}$. In (8), set $t = n\epsilon_n/4$. From the above calculations, $\mathbb{E}_{Y|X} \|G\|_{\mathcal{F}} \leq C_2 (n/\sigma_n^d)^{1/2} \leq n\epsilon_n/4$ since $t_1 \geq t_2 d/2$. Using $\sigma_{\mathcal{F}}^2 \leq C_1 n$, we finally obtain $\mathbb{P}_{Y|X}\Big(\|G\|_{\mathcal{F}} \geq n\epsilon_n/2 \mid X_{1:n} \Big) \leq 2\exp(-Cn\epsilon_n^2)$. ■

Lemma 5 *Under conditions of Theorem 1,*

$$\mathbb{P}_X \left(\left\| f_n^X - f_n \right\|_1 \geq \epsilon_n/2 \right) \leq \exp(-n\epsilon_n^2).$$

Proof As in Lemma 4, we express the desired probability in terms of a tail bound for the supremum of a stochastic process. However, the stochastic process in this case is no longer a Gaussian process and we cannot use Borell's inequality here. We instead use Bosquet's version of Talagrand's inequality for the supremum of a centered empirical process. The following Proposition 6 is adapted from Bousquet (2003) which also appears in Section 3.1 of Giné and Nickl (2011).

Proposition 6 *Assume $X_1, \ldots, X_n$ are independent and identically distributed as P. Let $\mathcal{G}$ be a countable set of real valued functions and assume all functions $g \in \mathcal{G}$ are P-measurable, square integrable and satisfy $\mathbb{E}_P[g] = 0$. Assume $K_1 = \sup_{g\in\mathcal{G}} \|g\|_\infty < \infty$ and let $W = \sup_{g\in\mathcal{G}} |\sum_{i=1}^n g(X_i)|$. Further, let $\sigma_{\mathcal{G}}^2 = \sup_{g\in\mathcal{G}} \mathbb{E}_P[g(X_1)^2]$ and $K_2 = n\sigma_{\mathcal{G}}^2 + K_1\mathbb{E}_P[W]$. Then, for any $t > 0$,*

$$\mathbb{P}\left\{W \geq \mathbb{E}_P W + (2K_2 t)^{1/2} + \frac{K_1 t}{3}\right\} \leq \exp(-t).$$

Let $L_x(t) = \psi_{\sigma_n}(x-t)f(t) - \psi_{\sigma_n} \star f(x)$ for $x, t \in [0,1]^d$ and $W = \int_{[0,1]^d} |\sum_{i=1}^n L_x(X_i)|dx$. Clearly, $\mathbb{P}_X(\|f_n^X - f_n\|_1 > \epsilon_n/2) = \mathbb{P}_X(W > n\epsilon_n/2)$. By an application of Hahn–Banach theorem as in the proof of Lemma 4, $W = \|G\|_{\mathcal{F}}$, where $\mathcal{F}$ is a countable dense subset of the unit ball of $L_\infty[0,1]^d$, $G(h) = \sum_{i=1}^n g(X_i)$, and $g(t) = \int_{[0,1]^d} L_x(t)h(x)dx$. Letting $\mathcal{G}$ denote the class of functions $\{g(t) = \int_{[0,1]^d} L_x(t)h(x)dx,\, h \in \mathcal{F}\}$, one has $\|G\|_{\mathcal{F}} = \sup_{g\in\mathcal{G}} |\sum_{i=1}^n g(X_i)|$. Putting together, $W = \sup_{g\in\mathcal{G}} |\sum_{i=1}^n g(X_i)|$ and $\mathbb{E}_X g(X_1) = 0$ by Tonelli's theorem. We now aim to apply Proposition 6 to bound $\mathbb{P}_X(W > n\epsilon_n/2)$. In order to apply Proposition 6, we need to estimate $K_1, \sigma_{\mathcal{G}}^2, K_2$ and $\mathbb{E}_P(W)$ which is carried out below.

Fix $g \in \mathcal{G}$. Then, there exists $h \in \mathcal{F}$ such that $g(t) = \int_{[0,1]^d} L_x(t)h(x)dx = f(t)\int_{[0,1]^d} \psi_{\sigma_n}(t-x)h(x)dx - \int \psi_{\sigma_n} \star f(x)\, h(x)dx$. Using the triangle inequality,

$$|g(t)| \leq |f(t)| \int_{[0,1]^d} |\psi_{\sigma_n}(t-x)h(x)|dx + \int_{[0,1]^d} |\psi_{\sigma_n} \star f(x)|\, |h(x)|dx.$$

Using $\|h\|_\infty \leq 1$, the first term in the above display can be bounded above by $C_1\|f\|_\infty$ where $C_1 = \int |\psi(t)|dt$. Similarly, the second term can be bounded above by $\|\psi_{\sigma_n} \star f\|_1 \leq \|\psi_{\sigma_n} \star f\|_\infty \leq \|f\|_\infty + \epsilon_n$, where the final inequality follows from (BS). Noting that for any $f \in \mathcal{P}_n$, $\|f\|_\infty \leq 2M_n$ (since the Hilbert space norm is stronger than the $\|\cdot\|_\infty$ norm), we have $K_1 \leq CM_n$.

Next we bound $\sigma_{\mathcal{G}}^2 = \sup_{g\in\mathcal{G}} \int_{[0,1]^d} g(t)^2 dt$. Fix $g \in \mathcal{G}$. Using the expression for $g(t)$ in the previous paragraph, $|g(t)| \leq |f(t)| \int |\psi_{\sigma_n}(x-t)|dx + \int |\psi_{\sigma_n} \star f(x)|dx$. As before, we can bound $\int |\psi_{\sigma_n}(x-t)|dx$ from above by C_1 and also $\int |\psi_{\sigma_n} \star f(x)|dx \leq C_1 \int_{s\in[0,1]^d} |f(s)|ds$. Using $(|a| + |b|)^2 \leq 2(|a|^2 + |b|^2)$ and the Cauchy–Schwarz inequality, $|g(t)|^2 \leq C|f(t)|^2 + C\{\int_{s\in[0,1]^d} |f(s)|ds\}^2 \leq C\{|f(t)|^2 + \|f\|_2^2\}$. Thus, we have $\sigma_{\mathcal{G}}^2 \leq C\|f\|_2^2$ for some absolute constant C. Using the bound for $\sup_{f\in\mathcal{P}_n} \|f\|_2^2$ in the following Proposition 7, we conclude that $\sigma_{\mathcal{G}}^2 \leq C$ for some absolute constant $C > 0$.

Proposition 7 *Recall $\mathcal{P}_n$ from* (5). *Then, $\sup_{f\in\mathcal{P}_n} \|f\|_2^2 \leq C$ for some absolute constant $C > 0$.*

Proof Let $f \in \mathcal{P}_n$. Then, there exists $h \in \mathbb{H}^{a_n}$ with $\|h\|_{\mathbb{H}^{a_n}} \leq M_n$ such that $\|f - h\|_\infty \leq \epsilon_n$. Hence, $\|f\|_2^2 \leq 2(\|h\|_2^2 + \epsilon_n^2)$ and it is enough to bound $\|h\|_2^2$. Recalling that $\|\cdot\|_{2,d}$ denotes the L_2 norm of $\mathbb{R}^d$, we have $\|h\|_2^2 \leq \|h\|_{2,d}^2$. We provide a bound for $\|h\|_{2,d}^2$ below.

There exists $\psi \in L_2(\mu_{a_n})$ such that $h(t) = \int \exp(i\langle\lambda, t\rangle)\xi(\lambda)\omega_{a_n}(\lambda)d\lambda$. Letting $\hat{h}$ denote the Fourier transform of h, one has from the Fourier inversion theorem that $\hat{h}(\lambda) =$

$\xi(-\lambda)\omega_{a_n}(\lambda)$. By Parseval's theorem, $\|h\|_{2,d}^2 = \|\hat{h}\|_{2,d}^2 = \int \xi^2(\lambda)\omega_{a_n}^2(\lambda)d\lambda$[1]. Observe that $\omega_{a_n}^2(\lambda) = a_n^{-2d}\exp\{-\|\lambda\|^2/(2a_n^2)\}/C^2$, where $C = 2^d\pi^{d/2}$. Hence,

$$\|h\|_{2,d}^2 = \frac{a_n^{-2d}}{C^2}\int \xi^2(\lambda)\exp\{-\|\lambda\|^2/(2a_n^2)\}d\lambda \leq \frac{a_n^{-2d}}{C^2}\int \xi^2(\lambda)\exp\{-\|\lambda\|^2/(4a_n^2)\}d\lambda$$
$$= \frac{a_n^{-d}}{C}\int \xi^2(\lambda)\omega_{a_n}(\lambda)d\lambda = \frac{\|h\|_{\mathbb{H}^{a_n}}^2}{Ca_n^d} \leq \frac{M_n^2}{Ca_n^d} = \frac{1}{C},$$

since $\|h\|_{\mathbb{H}^{a_n}}^2 = \left\|\xi\omega_{a_n}^{1/2}\right\|_{2,d}^2$ and $M_n = a_n^{d/2}$. ■

Finally, we proceed to bound $\mathbb{E}_X W$, where $W = \int_{[0,1]^d} |\sum_{i=1}^n L_x(X_i)|dx$. Using Jensen's inequality and the integral version of Minkowski's inequality, one has

$$\mathbb{E}_X W \leq (\mathbb{E}_X W^2)^{1/2} = \left[\int_{\prod_{i=1}^n [0,1]^d}\left\{\int_{[0,1]^d}|\sum_{i=1}^n L_x(t_i)dx|\right\}^2 dt_1\ldots dt_n\right]^{1/2}$$
$$\leq \int_{[0,1]^d}\left\{\int_{\prod_{i=1}^n [0,1]^d}|\sum_{i=1}^n L_x(t_i)|^2 dt_1\ldots dt_n\right\}^{1/2}dx.$$

Clearly, $\int_{\prod_{i=1}^n [0,1]^d}|\sum_{i=1}^n L_x(t_i)|^2 dt_1\ldots dt_n = \mathrm{Var}_X\{\sum_{i=1}^n L_x(X_i)\} = n\mathrm{Var}_X\{L_x(X_1)\}$, since $\mathbb{E}_X L_x(X_1) = 0$. Further, $\mathrm{Var}_X\{L_x(X_1)\} \leq \mathbb{E}_X\{\psi_{\sigma_n}(x - X_1)f(X_1)\}^2 = \int_{[0,1]^d}\psi_{\sigma_n}(x - t)^2 f(t)^2 dt \leq \frac{1}{\sigma_n^d}\psi_{\sigma_n} \star f^2$. Substituting this in the above display

$$\begin{aligned}
\mathbb{E}_X W &\leq \left(\frac{n}{\sigma_n^d}\right)^{1/2}\int_{[0,1]^d}\{\psi_{\sigma_n} \star f^2(x)\}^{1/2}dx \\
&\leq \left(\frac{n}{\sigma_n^d}\right)^{1/2}\left\{\int_{[0,1]^d}|\psi_{\sigma_n}| \star f^2(x)dx\right\}^{1/2} \\
&\leq \left(\frac{n}{\sigma_n^d}\right)^{1/2}\left\{\int_{[0,1]^d}\int_{[0,1]^d}|\psi_{\sigma_n}(x-t)|\, f^2(t)dtdx\right\}^{1/2} \\
&\leq \left(\frac{n}{\sigma_n^d}\right)^{1/2}\left[\int_{[0,1]^d} f^2(t)\left\{\int_{\mathbb{R}^d}|\psi_{\sigma_n}(x-t)|\,dx\right\}dt\right]^{1/2} \\
&\leq C\left(\frac{n}{\sigma_n^d}\right)^{1/2} = Cn^{\frac{\alpha+d}{2\alpha+d}}\log^{t_2 d/2} n \leq Cn\epsilon_n.
\end{aligned}$$

From the penultimate line to the last line of the above display, we invoked Proposition 7 to bound $\|f\|_2$ by a constant. We have thus obtained $K_1 \leq CM_n$ and $K_2 \leq Cn$. In Proposition 6, set $t = n\epsilon_n^2$. We have $K_1 t \leq C(n\epsilon_n M_n)\epsilon_n \leq n\epsilon_n$ for sufficiently large n provided $\alpha > d/2$. Further, $K_2 t \leq n^2\epsilon_n^2 + K_1\mathbb{E}_P(W)t = n^{\frac{2\alpha+2d}{2\alpha+d}}\log^{3t_1} n + n^{\frac{\alpha+2d+d/2}{2\alpha+d}}\log^{2t_1+t_2 d/2} n \leq 2n^{\frac{2\alpha+2d}{2\alpha+d}}\log^{3t_1} n$ for sufficiently large n if $\alpha > d/2$. Therefore, $(K_2 t)^{1/2} \leq n\epsilon_n$. ■

1. ω_{a_n} is symmetric about zero.

We next show that (BS) holds. Fix $f \in \mathcal{P}_n \cap U_n$. Since $f \in \mathcal{P}_n$, there exists $h \in \mathbb{H}^{a_n}$ with $\|h\|_{\mathbb{H}^{a_n}} \leq M_n$ such that $\|f - h\|_\infty \leq \epsilon_n$. By the triangle inequality, $\|\psi_{\sigma_n} \star f - f\|_1 \leq \|\psi_{\sigma_n} \star f - \psi_{\sigma_n} \star h\|_1 + \|\psi_{\sigma_n} \star h - h\|_1 + \|h - f\|_1$. Using $\|\psi_{\sigma_n} \star g\|_1 \leq \|g\|_1$ for any L_1 function g, we can further bound $\|\psi_{\sigma_n} \star f - f\|_1$ from above by $2\epsilon_n + \|\psi_{\sigma_n} \star h - h\|_\infty$. It thus remains to show that $\|\psi_{\sigma_n} \star h - h\|_\infty \leq \epsilon_n$.

There exists $\xi \in L_2(\mu_{a_n})$ such that $h(t) = \int \exp(i \langle \lambda, t \rangle) \xi(\lambda) \omega_{a_n}(\lambda) d\lambda$. Clearly, $\hat{h}(\lambda) = \xi(-\lambda)\omega_a(\lambda)$. Since the Fourier transform of $(\psi_{\sigma_n} \star h)$ is $(2\pi)^d \hat{\psi}_{\sigma_n} \hat{h}$ and $\hat{\psi}_{\sigma_n}(\lambda) = \hat{\psi}(\sigma_n \lambda)$, we have $\psi_{\sigma_n} \star h(t) = (2\pi)^d \int \exp(-i \langle \lambda, t \rangle) \hat{\psi}(\sigma_n \lambda) \hat{h}(\lambda) d\lambda$. We can choose ψ in a manner such that $\hat{\psi}$ is compactly supported, equals $(2\pi)^{-d}$ on $[-1, 1]^d$ and is bounded above by this constant everywhere; see proof of Lemma 4.3 in van der Vaart and van Zanten (2009). Putting together,

$$|\psi_{\sigma_n} \star h(t) - h(t)|^2 \leq \left\{ \int_{\|\lambda\| > \sigma_n^{-1}} |\hat{h}(\lambda)| \right\}^2 \leq \left\{ \int \xi(\lambda)^2 \omega_{a_n}(\lambda) d\lambda \right\} \int_{\|\lambda\| > \sigma_n^{-1}} \omega_{a_n}(\lambda) d\lambda$$
$$\leq C \|h\|^2_{\mathbb{H}^{a_n}} \exp\{-\sigma_n^{-2}/(4a_n^2)\} \leq C M_n^2 \exp\{-\sigma_n^{-2}/(4a_n^2)\} = C a_n^d \exp\{-(\log^{2t_2} n/4)\},$$

where C is an absolute constant. The proof follows by noting that $C a_n^d \exp\{-\log^{2t_2} n/4\} \leq \epsilon_n^2$ whenever $t_2 > 1/2$ (holds for $t_2 = 1/(2-\kappa)$, for $0 < \kappa < 1$).

3.2 Proof of Corollary 2

Case $\beta < \alpha$: Setting $\sigma_n = n^{-1/(2\beta+d)} \log^{-t_2} n$ for some constant $t_2 \geq 1/(2-\kappa)$, for $0 < \kappa < 2$, $M_n = a_n^{d/2}$, $\tilde{f}_n$ same as in (4), $\mathcal{P}_n = M_n \mathbb{H}_1^{a_n} + \epsilon_n \mathbb{B}_1$ with $\epsilon_n = n^{-\beta/(2\beta+d)} \log^{3t_1/2} n$, $t_1 \geq t_2 d/2$, and $\delta_n = \epsilon_n = \epsilon_n'$, one can verify (PCS), (BT), (DT), (BS), (DS) exactly as in the proof of Theorem 1. (PCN) follows from Lemma 4.3 of van der Vaart and van Zanten (2009).
Case $\beta > \alpha$: Same as before with $\epsilon_n = n^{-\alpha/(2\beta+d)} \log^{3t_1/2} n$ for $t_1 \geq t_2 d/2$.

4. Discussion

The article extends upon previous results on random design regression using Gaussian process priors. A limitation of the current exposition is the requirement of the knowledge of the smoothness parameter to construct the rescaling sequence. A natural question is whether one can find a suitable prior on the bandwidth parameter which adapts to the unknown smoothness level as in the fixed design case in van der Vaart and van Zanten (2009). We propose to resolve this issue as a part of future research. Also, our current proof technique would lead to a sub-optimal rate of posterior convergence for L_p norms with $p \neq 1$. We believe this is due to the use of Talagrand's inequality to construct the test function. A key requirement to obtain optimal convergence rate is that the *variance term* $\sigma_{\mathcal{F}}^2$ in the application of Talagrand's inequality should be at most $O(n)$. This assertion is true only when $p = 1$. Obtaining convergence rates for integrated L_p norms with $p \neq 1$ is a topic of future research.

Acknowledgement

Dr. Pati and Dr. Bhattacharya acknowledge support for this project from the Office of Naval Research (ONR BAA 14-0001). Dr. Cheng acknowledges support from the National Science Foundation (NSF CAREER, DMS – 1151692, DMS – 1418042), Simons Foundation (305266) and warm hosting at SAMSI.

Appendix A. Proof of Theorem 3

Let $\|f\|_{2,n}$ denote the empirical L_2 norm of f, so that $\|f\|_{2,n}^2 = n^{-1}\sum_{i=1}^n f^2(X_i)$. Also, define

$$L_n(f, f_0) = \frac{q^{(n)}(Y_{1:n}, X_{1:n} \mid f)}{q^{(n)}(Y_{1:n}, X_{1:n} \mid f_0)}.$$

Lemma 8 *Let A_n denote the following event in the sigma-field generated by $(Y_{1:n}, X_{1:n})$:*

$$A_n = \left\{(Y_{1:n}, X_{1:n}) : \int L_n(f, f_0)\Pi(df) \geq e^{-n\delta_n^2}\Pi(\|f - f_0\|_\infty \leq \delta_n)\right\}. \tag{11}$$

Then, $\mathbb{P}_{Y,X}^{f_0}(A_n) \geq 1 - e^{-Cn\delta_n^2}$.

Proof Clearly, $\mathbb{P}_{Y,X}^{f_0}(A_n) = \mathbb{E}_X^{f_0}[\mathbb{P}_{Y|X}^{f_0}(A_n)]$. By Lemma 14 of van der Vaart and van Zanten (2011), $\mathbb{P}_{Y|X}^{f_0}\{\int L_n(f, f_0)\Pi(df) \geq e^{-n\delta_n^2}\Pi(\|f - f_0\|_{2,n} \leq \delta_n)\} \geq 1 - e^{-n\delta_n^2/8}$. The conclusion follows by noting that $\Pi(\|f - f_0\|_\infty < \delta_n) \leq \Pi(\|f - f_0\|_{2,n} < \delta_n)$. ■

Lemma 9 *There exists a test function Φ_n for $H_0 : f = f_0$ vs $H_1 : f \in U_n \cap \mathcal{P}_n$ such that*

$$\mathbb{E}_{Y,X}^{f_0}\Phi_n \leq e^{-Cn\delta_n^2}, \tag{12}$$

$$\sup_{f \in U_n \cap \mathcal{P}_n} \mathbb{E}_{Y,X}^{f}(1 - \Phi_n) \leq e^{-Cn\delta_n^2}. \tag{13}$$

for some absolute constant C.

Proof Let $\Phi_n = 1(\|\tilde{f}_n - f_0\|_1 > M\epsilon_n/2)$. The error bounds follow from (BT), (DT) and (BS), (DS). ■

Using a standard line of argument for establishing convergence rates in Bayesian non-parametric models (Ghosal et al., 2000), we have $\mathbb{E}_{Y,X}^{f_0}\Pi(U_n \mid Y_{1:n}, X_{1:n}) \leq \sum_{i=1}^4 b_{in}$, where $b_{1n} = \mathbb{E}_{Y,X}^{f_0}\Phi_n$, $b_{2n} = e^{n\delta_n^2}\sup_{f \in U_n \cap \mathcal{P}_n}\mathbb{E}_{Y,X}^{f}(1 - \Phi_n)/\Pi(\|f - f_0\|_\infty < \delta_n)$, $b_{3n} = e^{n\delta_n^2}\Pi(\mathcal{P}_n^c)/\Pi(\|f - f_0\|_\infty < \delta_n)$ and $b_{4n} = \mathbb{P}_{Y,X}^{f_0}(A_n^c)$. The Theorem then follows from Lemmas 8, 9 and Conditions (PCS) and (PCN).

References

R. J. Adler. *An Introduction to Continuity, Extrema, and Related Topics for General Gaussian processes*, volume 12. Institute of Mathematical Statistics, 1990.

Y. Baraud. Model selection for regression on a random design. *ESAIM: Probability and Statistics*, 6:127–146, 2002.

A. Bhattacharya, D. Pati, and D. B. Dunson. Anisotropic function estimation using multi-bandwidth gaussian processes. *The Annals of Statistics*, 42(1):352–381, 2014.

L. Birgé. Sur un theorémè de minimax et son application aux tests. *Probability and Mathematical Statistics*, 3:259–282, 1984.

L. Birgé. Model selection for gaussian regression with random design. *Bernoulli*, 10(6): 1039–1051, 2004.

D. Bontemps. Bernstein-von Mises theorems for Gaussian regression with increasing number of regressors. *The Annals of Statistics*, 39(5):2557–2584, 2011.

O. Bousquet. Concentration inequalities for sub-additive functions using the entropy method. *Progress in Probability*, pages 213–248, 2003.

L. D. Brown, T. T. Cai, M. G. Low, and C-H. Zhang. Asymptotic equivalence theory for nonparametric regression with random design. *The Annals of Statistics*, pages 688–707, 2002.

T. Choi and M.J. Schervish. On posterior consistency in nonparametric regression problems. *Journal of Multivariate Analysis*, 98(10):1969–1987, 2007.

S. Ghosal and A. Roy. Posterior consistency of Gaussian process prior for nonparametric binary regression. *The Annals of Statistics*, 34(5):2413–2429, 2006.

S. Ghosal, J. K. Ghosh, and A. W. van der Vaart. Convergence rates of posterior distributions. *The Annals of Statistics*, 28(2):500–531, 2000.

E. Giné and R. Nickl. Rates of contraction for posterior distributions in L^r-metrics, $1 \leq r \leq \infty$. *The Annals of Statistics*, 39(6):2883–2911, 2011.

B. J. K. Kleijn and A. W. van der Vaart. Misspecification in infinite-dimensional Bayesian statistics . *The Annals of Statistics*, 34(2):837–877, 2006.

L. Le Cam. Asymptotic methods in statistical decision theory. *New York*, 1986.

D. Pati, A. Bhattacharya, N. S. Pillai, and D. B. Dunson. Posterior contraction in sparse Bayesian factor models for massive covariance matrices. *The Annals of Statistics*, 42(3): 1102–1130, 2014.

C. E. Rasmussen. Gaussian processes in machine learning. *Advanced Lectures on Machine Learning*, pages 63–71, 2004.

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

K. Ray. Bayesian inverse problems with non-conjugate priors. *Electronic Journal of Statistics*, 7:2516–2549, 2013.

M. Seeger. Gaussian processes for machine learning. *International Journal of Neural Systems*, 14(02):69–106, 2004.

M. Seeger, S. M. Kakade, and D. P. Foster. Information consistency of nonparametric gaussian process methods. *IEEE Transactions on Information Theory*, 54(5):2376–2382, 2008.

Z. Shang and G. Cheng. Nonparametric Bernstein-von Mises phenomenon: A tuning prior perspective. *ArXiv Preprint ArXiv:1411.3686*, 2014.

B. T. Szabó, A. W. van der Vaart, and J. H. van Zanten. Empirical bayes scaling of gaussian priors in the white noise model. *Electronic Journal of Statistics*, 7:991–1018, 2013.

S. T. Tokdar and J. K. Ghosh. Posterior consistency of logistic Gaussian process priors in density estimation. *Journal of Statistical Planning and Inference*, 137(1):34–42, 2007.

A. W. van der Vaart and J. H. van Zanten. Bayesian inference with rescaled Gaussian process priors. *Electronic Journal of Statistics*, 1:433–448, 2007. ISSN 1935-7524.

A. W. van der Vaart and J. H. van Zanten. Rates of contraction of posterior distributions based on gaussian process priors. *The Annals of Statistics*, 36(3):1435–1463, 2008a.

A. W. van der Vaart and J. H. van Zanten. Reproducing kernel Hilbert spaces of Gaussian priors. *IMS Collections*, 3:200–222, 2008b.

A. W. van der Vaart and J. H. van Zanten. Adaptive Bayesian estimation using a Gaussian random field with inverse Gamma bandwidth. *The Annals of Statistics*, 37(5B):2655–2675, 2009.

A. W. van der Vaart and J. H. van Zanten. Information rates of nonparametric Gaussian process methods. *Journal of Machine Learning Research*, 12:2095–2119, 2011.

Journal of Machine Learning Research 16 (2015) 2853-2858 Submitted 9/14; Revised 2/15; Published 12/15

CEKA: A Tool for Mining the Wisdom of Crowds

Jing Zhang JZHANG@NJUST.EDU.CN
School of Computer Science and Information Engineering
Hefei University of Technology (HFUT), Hefei 230009, China
Department of Software Engineering, School of Computer Science and Engineering
Nanjing University of Science and Technology (NJUST), Nanjing 210094, China

Victor S. Sheng SSHENG@UCA.EDU
Bryce A. Nicholson BNICHOLSON1210@HOTMAIL.COM
Department of Computer Science, University of Central Arkansas, Conway, AR 72035, USA

Xindong Wu XWU@UVM.EDU
School of Computer Science and Information Engineering, HFUT, Hefei 230009, China
Department of Computer Science, University of Vermont, Burlington, VT 05405, USA

Editor: Mark Reid

Abstract

CEKA is a software package for developers and researchers to mine the wisdom of crowds. It makes the entire knowledge discovery procedure much easier, including analyzing qualities of workers, simulating labeling behaviors, inferring true class labels of instances, filtering and correcting mislabeled instances (noise), building learning models and evaluating them. It integrates a set of state-of-the-art inference algorithms, a set of general noise handling algorithms, and abundant functions for model training and evaluation. CEKA is written in Java with core classes being compatible with the well-known machine learning tool WEKA, which makes the utilization of the functions in WEKA much easier.

Keywords: crowdsourcing, learning from crowds, multiple noisy labeling, inference, noise handling, repeated labeling simulation

1. Introduction

The emergence of crowdsourcing (Howe, 2006) has changed the way of knowledge acquisition. It has already attracted vast attentions of the machine learning and data mining research community in the past several years. Researchers show great interests in utilizing crowdsourcing as a new approach to acquire class labels of objects from common users, which costs much less than the traditional way—annotating by domain experts. In order to improve the labeling quality, an object usually obtains multiple labels from different non-expert annotators. Then, inference algorithms will be introduced to estimate the ground truths of these objects. Many inference algorithms have been proposed in recent years. Besides, building learning models from the inferred crowdsourced data is another research issue with great challenges, which aims at lifting the quality of a learned model to the level that can be achieved by training with the data labeled by domain experts.

To facilitate the research on mining the wisdom of crowds, we develop a novel software package named Crowd Environment and its Knowledge Analysis (CEKA). The main contri-

bution of CEKA lies on three aspects. (1) It provides comprehensive functions, which not only includes a great number of ground truth inference algorithms with a uniform easy-to-use programming interface but also includes a lot of well designed functions for the management of crowdsourced data. (2) It is seamlessly compatible with the famous machine learning tool WEKA (Hall et al., 2009), which facilitates the combination of the previous inference and the subsequent model learning procedures. (3) It is written in Java and completely open source. Therefore, many new ideas and methods, such as noise correction for crowdsourcing, are easily integrated. The project CEKA is available at: http://ceka.sourceforge.net/.

2. Design Principles and System Architecture

The design of CEKA follows three basic guidelines. (1) *Preferring integration of existing algorithms rather than implementing them.* Unless the original implementations of algorithms are not released, we always try to integrate the original versions rather than re-implementing them. The work that we have done is to unify the input/output file formats and wrap the different algorithms into some newly designed java classes with a uniform easy-to-use member functions. (2) *Seamlessly compatible with WEKA.* When input files that contain crowdsourced data are loaded into the memory and form a `Dataset` object, this object `Dataset` and all `Examples` inside can directly cooperate (e.g. training a model and conducting a cross-validation) with the related classes in WEKA. (3) *Extendibility.* Because machine learning in crowdsourcing is an emerging research domain, many topics such as multi-label tasks in crowdsourcing have not been touched yet. In order to integrate future research easily, when designing the core components of CEKA, we attempt to make the class structures as extendable as possible.

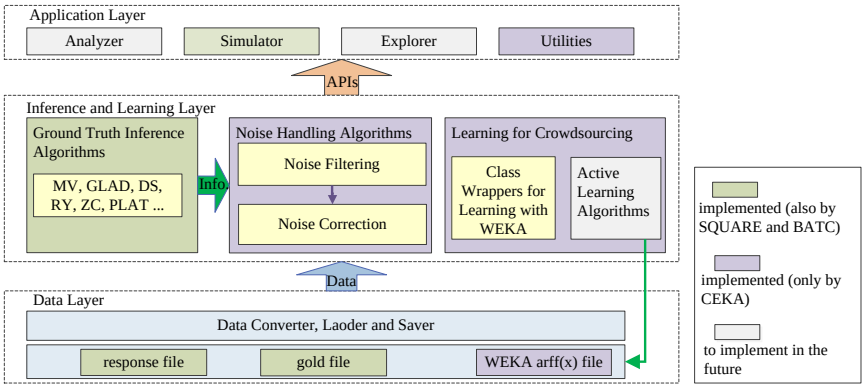


Figure 1: The architecture of CEKA

Figure 1 illustrates the hierarchical architecture of CEKA, in which we also compare it with two other tools for crowdsourcing SQUARE (Sheshadri and Lease, 2013) and BATC (Nguyen et al., 2013). Generally, SQUARE and BATC only provide some inference algorithms and several simple analysis functions. By contrast, CEKA conceives a more ambitious blueprint. It attempts to support the entire knowledge discovery procedure including analysis, inference and model learning. In the data layer, CEKA is able to read an arff(x) file defined by WEKA, which contains features of instances for subsequent model building. In the inference and learning layer, it provides a large number of inference algorithms. Our

on-going studies find that mislabeled instances after inference can be effectively detected and corrected, if a noise (mislabeled instance) handling algorithm can take advantage of the information generated in the previous inference procedure. Thus, CEKA provides a batch of noise handling algorithms. The core classes in this layer are derived from related classes in WEKA. In the application layer, CEKA provides a lot of utilities such as calculating performance evaluation metrics (i.e., accuracy, recall, precision, F source, AUC, M-AUC), manipulating data (i.e., shuffling, splitting and combining data), etc.

Algo.	CEKA	SQUARE	BATC	Comments	Algo.	CEKA	SQUARE	BATC	Comments
MV	●	●	●		CF	●			
DS	●	●	●		IPF	●			
GLAD	●	●	●	transplanted to Windows	MPF	●			
KOS	●		●		VF	●			
RY	●	●	●	by SQUARE	PLC	●			
ZenCrowd	●	●		by SQUARE	STC	●			
PLAT	●			for biased binary labeling	CC	●			unpublished
AWMV	●			unpublished					
GTIC	●			unpublished					

Table 1: Algorithms in CEKA compared with SQUARE and BATC

3. Algorithms

For the anonymous nature of crowdsourcing, CEKA currently only focuses on agnostic inference algorithms, which are independent of any other prior knowledge besides annotations assigned by non-experts. CEKA includes several novel inference algorithms proposed by the authors such as ground truth inference using clustering (GTIC) for multi-class labeling, adaptive weighted majority Voting (AWMV) for biased binary labeling as well as the well-known algorithms majority voting (MV), Dawid & Skene's algorithm (DS) (Dawid and Skene, 1979), GLAD (Whitehill et al., 2009), KOS (Karger et al., 2011), RY (Raykar et al., 2010), ZenCrowd (Demartini et al., 2012), and PLAT (Zhang et al., 2015). To embody our thought of introducing noise handling to improve the data quality of crowdsourcing, we have proposed a novel framework and an algorithm adaptive voting noise correction (AVNC) for crowdsourcing. In this framework, CEKA also includes a batch of noise filtering and correction algorithms, such as classification filtering (CF) (Gamberger et al., 1999), iterative partition filtering (IPF) (Khoshgoftaar and Rebours, 2007), multiple partition filtering (MPF) (Khoshgoftaar and Rebours, 2007), voting filtering (VF) (Brodley and Friedl, 1999), polishing label correction (PLC) (Teng, 1999), self-training correction (STC) (Triguero et al., 2014) and clustering correction (CC). Table 1 lists all algorithms in its current version (v1.0), comparing with SQUARE (Sheshadri and Lease, 2013) and BATC (Nguyen et al., 2013). Although our proposed algorithms GTIC, AWMV, and CC are under review, all of them still can be accessed in the source code.

4. Usage Example

CEKA can be easily deployed in both Windows and Linux systems. We have transplanted some algorithms such as GLAD from Linux to Windows. Figure 2 demonstrates a simple ex-

periment including the ground truth inference, noise correction and performance evaluation. In this sample code, like DS, all inference algorithms provide a uniform interface function `doInference`, which assigns every instance an integrated label. The class `Dataset` is completely compatible with the class `Instances` in WEKA, which can be directly accepted by a WEKA classifier as its parameter to train a model. Simply as the code shows, the statistical information of the performance will be obtained when the class `PerformanceStatistic` is applied to a `Dataset` object with the ground truth provided.

```
String respPath=D:/adult.response.txt; // labels obtained from crowd
String arffPath=D:/adult.arffx;        // ground truth and features
Dataset data = loadFile(respPath, null, arffPath);
// infer the ground truth by Dawid & Skene's algorithm
DawidSkene dsAlgo = new DawidSkene(50);
dsAlgo.doInference(data);
// noise filtering with the CF algorithm
Classifier [] classifiers = new Classifier[1];
Classifiers[0] = new SMO();            // SMO Classifier in WEKA
ClassificationFilter noiseFilter = new ClassificationFilter(10);
Dataset[] subData = null;              // cleansed and noise data sets
cf.FilterNoise(data, classifiers[0]);  // conduct noise filtering
subData[0] = noiseFilter.getCleansedDataset();
subData[1] = noiseFilter.getNoiseDataset();
// noise correction with STC algorithm
SelfTrainCorrection stc = new SelfTrainCorrection(subData[0], subData[1], 1.0);
stc.correction(classifiers[0]);        // correct mislabeled data
// combining two data sets and then evaluate performance
DatasetManipulator.addAllExamples(subData[0], subData[1]);
PerformanceStatistic perfStat = new PerformanceStatistic();
perfStat.stat(subData[0]);
```

Figure 2: A sample code for a basic usage

5. Conclusion and Future Work

CEKA is an easy-to-use open-source package for inference and machine learning tasks in crowdsourcing. The current version of CEKA includes a large number of ground truth inference algorithms, noise handling algorithms and useful functions supporting different learning tasks. That CEKA is designed to cooperate with WEKA definitely facilitates and accelerates the research progress in this field. CEKA is still growing. The future work includes introducing crowdsourcing-specific active learning strategies, developing several GUI tools (analyzer, simulator, and explorer) as well as integrating more inference, noise handling and learning algorithms proposed either by the authors or other researchers.

Acknowledgments

This research has been supported by the Program for Changjiang Scholars and Innovative Research Team in University (PCSIRT) of the Ministry of Education, China, under grant IRT13059, the National 973 Program of China under grant 2013CB329604, the National

Natural Science Foundation of China under grant 61229301, and the US National Science Foundation under grant IIS-1115417.

References

Carla E Brodley and Mark A Friedl. Identifying mislabeled training data. *Journal of Artificial Intelligence Research*, 11:131–161, 1999.

Alexander Philip Dawid and Allan M Skene. Maximum likelihood estimation of observer error-rates using the em algorithm. *Applied Statistics*, pages 20–28, 1979.

Gianluca Demartini, Djellel Eddine Difallah, and Philippe Cudré-Mauroux. Zencrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In *World Wide Web*, pages 469–478. ACM, 2012.

Dragan Gamberger, Nada Lavrac, and Ciril Groselj. Experiments with noise filtering in a medical domain. In *ICML*, pages 143–151, 1999.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The weka data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, 2009.

Jeff Howe. The rise of crowdsourcing. *Wired Magazine*, 14(6):1–4, 2006.

David R Karger, Sewoong Oh, and Devavrat Shah. Iterative learning for reliable crowdsourcing systems. In *NIPS*, pages 1953–1961, 2011.

Taghi M Khoshgoftaar and Pierre Rebours. Improving software quality prediction by noise filtering techniques. *Journal of Computer Science and Technology*, 22(3):387–396, 2007.

Quoc Viet Hung Nguyen, Thanh Tam Nguyen, Ngoc Tran Lam, and Karl Aberer. Batc: a benchmark for aggregation techniques in crowdsourcing. In *ACM SIGIR*, pages 1079–1080. ACM, 2013.

Vikas C Raykar, Shipeng Yu, Linda H Zhao, Gerardo Hermosillo Valadez, Charles Florin, Luca Bogoni, and Linda Moy. Learning from crowds. *Journal of Machine Learning Research*, 11:1297–1322, 2010.

Aashish Sheshadri and Matthew Lease. Square:. In *The First AAAI Conference on Human Computation and Crowdsourcing*, pages 156–164, 2013.

Choh-Man Teng. Correcting noisy data. In *ICML*, pages 239–248, 1999.

Isaac Triguero, José A Sáez, Julián Luengo, Salvador García, and Francisco Herrera. On the characterization of noise filters for self-training semi-supervised in nearest neighbor classification. *Neurocomputing*, 132:30–41, 2014.

Jacob Whitehill, Ting-fan Wu, Jacob Bergsma, Javier R Movellan, and Paul L Ruvolo. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *NIPS*, pages 2035–2043, 2009.

Jing Zhang, Xindong Wu, and Victor S Sheng. Imbalanced multiple noisy labeling. *IEEE Transaction on Kownledge and Data Engineering*, 27(2):489–503, 2015.

Journal of Machine Learning Research 16 (2015) 2859-2900 Submitted 5/14; Revised 3/15; Published 12/15

Linear Dimensionality Reduction: Survey, Insights, and Generalizations

John P. Cunningham JPC2181@COLUMBIA.EDU
Department of Statistics
Columbia University
New York City, USA

Zoubin Ghahramani ZOUBIN@ENG.CAM.AC.UK
Department of Engineering
University of Cambridge
Cambridge, UK

Editor: Gert Lanckriet

Abstract

Linear dimensionality reduction methods are a cornerstone of analyzing high dimensional data, due to their simple geometric interpretations and typically attractive computational properties. These methods capture many data features of interest, such as covariance, dynamical structure, correlation between data sets, input-output relationships, and margin between data classes. Methods have been developed with a variety of names and motivations in many fields, and perhaps as a result the connections between all these methods have not been highlighted. Here we survey methods from this disparate literature as optimization programs over matrix manifolds. We discuss principal component analysis, factor analysis, linear multidimensional scaling, Fisher's linear discriminant analysis, canonical correlations analysis, maximum autocorrelation factors, slow feature analysis, sufficient dimensionality reduction, undercomplete independent component analysis, linear regression, distance metric learning, and more. This optimization framework gives insight to some rarely discussed shortcomings of well-known methods, such as the suboptimality of certain eigenvector solutions. Modern techniques for optimization over matrix manifolds enable a generic linear dimensionality reduction solver, which accepts as input data and an objective to be optimized, and returns, as output, an optimal low-dimensional projection of the data. This simple optimization framework further allows straightforward generalizations and novel variants of classical methods, which we demonstrate here by creating an orthogonal-projection canonical correlations analysis. More broadly, this survey and generic solver suggest that linear dimensionality reduction can move toward becoming a blackbox, objective-agnostic numerical technology.

Keywords: dimensionality reduction, eigenvector problems, matrix manifolds

1. Introduction

Linear dimensionality reduction methods have been developed throughout statistics, machine learning, and applied fields for over a century, and these methods have become indispensable tools for analyzing high dimensional, noisy data. These methods produce a

low-dimensional linear mapping of the original high-dimensional data that preserves some feature of interest in the data. Accordingly, linear dimensionality reduction can be used for visualizing or exploring structure in data, denoising or compressing data, extracting meaningful feature spaces, and more. This abundance of methods, across a variety of data types and fields, suggests a great complexity to the space of linear dimensionality reduction techniques. As such, there has been little effort to consolidate our understanding. Here we survey a host of methods and investigate when a more general optimization framework can improve performance and extend the generality of these techniques.

We begin by defining linear dimensionality reduction (Section 2), giving a few canonical examples to clarify the definition. We then interpret linear dimensionality reduction in a simple optimization framework as a program with a problem-specific objective over orthogonal or unconstrained matrices. Section 3 surveys principal component analysis (PCA; Pearson, 1901; Eckart and Young, 1936), multidimensional scaling (MDS; Torgerson, 1952; Cox and Cox, 2001; Borg and Groenen, 2005), Fisher's linear discriminant analysis (LDA; Fisher, 1936; Rao, 1948), canonical correlations analysis (CCA; Hotelling, 1936), maximum autocorrelation factors (MAF; Switzer and Green, 1984), slow feature analysis (SFA; Wiskott and Sejnowski, 2002; Wiskott, 2003), sufficient dimensionality reduction (SDR; Fukumizu et al., 2004; Adragni and Cook, 2009), locality preserving projections (LPP; He and Niyogi, 2004; He et al., 2005), undercomplete independent component analysis (ICA; e.g. Hyvarinen et al., 2001), linear regression, distance metric learning (DML; Kulis, 2012; Yang and Jin, 2006), probabilistic PCA (PPCA; Tipping and Bishop, 1999; Roweis, 1997; Theobald, 1975), factor analysis (FA; Spearman, 1904), several related methods, and important extensions such as kernel mappings and regularizations.

A common misconception is that many or all linear dimensionality reduction problems can be reduced to eigenvalue or generalized eigenvalue problems. Not only is this untrue in general, but it is also untrue for some very well-known algorithms that are typically thought of as generalized eigenvalue problems. The suboptimality of using eigenvector bases in these settings is rarely discussed and is one notable insight of this survey. Perhaps inherited from this eigenvalue misconception, a second common tendency is for practitioners to greedily choose the low-dimensional data: the first dimension is chosen to optimize the problem objective, and then subsequent dimensions are chosen to optimize the objective on a residual or reduced data set. The optimization framework herein shows the limitation of this view. More importantly, the framework also suggests a more generalized linear dimensionality reduction solver that encompasses all eigenvalue problems as well as many other important variants. In this survey we restate these algorithms as optimization programs over matrix manifolds that have a well understood geometry and a well developed optimization literature (Absil et al., 2008). This simple perspective leads to a generic algorithm for linear dimensionality reduction, suggesting that, like numerical optimization more generally, linear dimensionality reduction can become abstracted as a numerical technology for a range of problem-specific objectives. In all, this work: *(i)* surveys the literature on linear dimensionality reduction, *(ii)* gives insights to some rarely discussed shortcomings of traditional approaches, and *(iii)* provides a simple algorithmic template for generalizing to many more problem-specific techniques.

2. Linear Dimensionality Reduction as a Matrix Optimization Program

We define linear dimensionality reduction as all methods with the problem statement:

Definition 1 (Linear Dimensionality Reduction) *Given n d-dimensional data points $X = [x_1, ..., x_n] \in \mathbb{R}^{d \times n}$ and a choice of dimensionality $r < d$, optimize some objective $f_X(\cdot)$ to produce a linear transformation $P \in \mathbb{R}^{r \times d}$, and call $Y = PX \in \mathbb{R}^{r \times n}$ the low-dimensional transformed data.*

Note that throughout this work we assume without loss of generality that data X is mean-centered, namely $X1 = 0$. To make this definition concrete, we briefly detail two widespread linear dimensionality reduction techniques: principal component analysis (PCA; Pearson, 1901) and canonical correlations analysis (CCA; Hotelling, 1936). PCA maximizes data variance captured by the low-dimensional projection, or equivalently minimizes the reconstruction error (under the ℓ_2-norm) of the projected data points with the original data, namely

$$f_X(M) = ||X - MM^\top X||_F^2.$$

Here M is a matrix with r orthonormal columns. In the context of Definition 1, optimizing $f_X(M)$ produces an M such that $P = M^\top$, and the desired low-dimensional projection is $Y = M^\top X$. PCA is discussed in depth in Section 3.1.1.

We stress that the notation of M and P in Definition 1 is not redundant, but rather is required for other linear dimensionality reduction techniques where the linear transformation P does not equal the optimization variable M (as it does in PCA). Consider CCA, another classical linear dimensionality reduction technique that jointly maps two data sets $X_a \in \mathbb{R}^{d_a \times n}$ and $X_b \in \mathbb{R}^{d_b \times n}$ to $Y_a \in \mathbb{R}^{r \times n}$ and $Y_b \in \mathbb{R}^{r \times n}$, such that the sample correlation between Y_a and Y_b is maximized[1]. Under the additional constraints that Y_a and Y_b have uncorrelated variables ($Y_a Y_b^\top = \Lambda$, a diagonal matrix) and be individually uncorrelated with unit variance ($\frac{1}{n} Y_a Y_a^\top = \frac{1}{n} Y_b Y_b^\top = I$), a series of standard steps produces the well known objective

$$f_X(M_a, M_b) = \frac{1}{r} \mathrm{tr}\left(M_a^\top (X_a X_a^\top)^{-1/2} X_a X_b^\top (X_b X_b^\top)^{-1/2} M_b \right),$$

as will be detailed in depth in Section 3.1.4. This objective is maximized when $M_a^\top$ and $M_b^\top$ are the left and right singular vectors of the matrix $(X_a X_a^\top)^{-1/2} X_a X_b^\top (X_b X_b^\top)^{-1/2}$. In the context of Definition 1, the low dimensional canonical variables Y_a are then related to the original data as $Y_a = P_a X_a \in \mathbb{R}^{r \times n}$, where $P_a = M_a^\top \left(X_a X_a^\top \right)^{-1/2}$ (and similar for Y_b). Since M_a has by definition orthonormal columns, CCA, by inclusion of the whitening term $\left(X_a X_a^\top \right)^{-1/2}$, does not represent an orthogonal projection of the data. Accordingly, CCA and PCA point out two key features of linear dimensionality reduction and Definition 1: first, that the objective function $f_X(\cdot)$ need not entirely define the linear mapping P to the low-dimensional space; and second, that not all linear dimensionality reduction methods need be orthogonal projections, or indeed projections at all.

1. As a point of technical detail, note that the use of two data sets and mappings is only a notational convenience; writing the CCA projection as $Y = \begin{bmatrix} Y_a \\ Y_b \end{bmatrix} = \begin{bmatrix} P_a & 0 \\ 0 & P_b \end{bmatrix} \begin{bmatrix} X_a \\ X_b \end{bmatrix} = PX$, we see that CCA adheres precisely to Definition 1.

Note also that both PCA and CCA result in a matrix decomposition, and indeed a common approach to many linear dimensionality reduction methods is to attempt to cast the problem as an eigenvalue or generalized eigenvalue problem (Burges, 2010). This pursuit can be fruitful but is limited, often resulting in ad hoc or suboptimal algorithms. As a specific example, in many settings orthogonal projections of data are required for visualization and other basic needs. Can we create an *Orthogonal CCA*, where we seek orthogonal projections $Y_a = M_a^\top X_a$ for a matrix M_a with orthonormal columns (and similar for Y_b), such that the sample correlation between Y_a and Y_b is maximized? No known eigenvalue problem can produce this projection, so one tempting and common approach is to orthonormalize P_a and P_b (the results found by traditional CCA). We will show that this choice can be significantly suboptimal, and in later sections we will create Orthogonal CCA using a generic optimization program. Thus matrix decomposition approaches suggest an unfortunate limitation to the set of possible linear dimensionality reduction problems, and a broader framework is required to fully capture Definition 1 and linear dimensionality reduction.

2.1 Optimization Framework for Linear Dimensionality Reduction

All linear dimensionality reduction methods presented here can be viewed as solving an optimization program over a matrix manifold $\mathcal{M}$, namely

$$\begin{aligned} &\text{minimize} && f_X(M) \\ &\text{subject to} && M \in \mathcal{M}. \end{aligned} \tag{1}$$

Given Definition 1, the intuition behind this optimization program should be apparent: the objective $f_X(\cdot)$ defines the feature of interest to be captured in the data, and the matrix manifold encodes some aspects of the linear mapping P such that $Y = PX$.[2]

All methods considered here specify M as one of two matrix forms. First, some methods are unconstrained optimizations over rank r linear mappings, implying the trivial manifold constraint of Euclidean space, which we denote as $M \in \mathbb{R}^{d\times r}$. In this case, optimization may be straightforward, and algorithms like expectation-maximization (Dempster et al., 1977) or standard first order solvers have been well used.

Second, very often the matrix form will have an orthogonality constraint $\mathcal{M} = \{M \in \mathbb{R}^{d\times r} \ : \ M^\top M = I\}$, corresponding to orthogonal projections of the data X. In this case we write $\mathcal{M} = \mathcal{O}^{d\times r}$. As noted previously, the typical and often flawed approach is to attempt to cast these problems as eigenvalue problems. Instead, viewed through the lens of Equation 1, linear dimensionality reduction is simply an optimization program over a matrix manifold, and indeed there is a well-developed optimization literature for matrix manifolds (foundations include Luenberger, 1972; Gabay, 1982; Edelman et al., 1998; an excellent summary is Absil et al., 2008).

As a primary purpose of this work is to survey linear dimensionality reduction, we first detail linear dimensionality reduction techniques using this optimization framework. We then implement a generic solver for programs of the form Equation 1, where $\mathcal{M}$ is the family of orthogonal matrices $\mathcal{O}^{d\times r}$. Thus we show the framework of Equation 1 to be not

2. Note that several methods will require optimization over additional auxiliary unconstrained variables, which can be addressed algorithmically via a coordinate descent approach (alternating optimizations over the auxiliary variable and Equation 1) or some more nuanced scheme.

only conceptually simplifying, but also algorithmically simplifying. Instead of resorting to ad hoc (and often suboptimal) formulations for each new problem in linear dimensionality reduction, practitioners need only specify the objective $f_X(\cdot)$ and the high-dimensional data X, and these numerical technologies can produce the desired low-dimensional data. Section 4 validates this claim by applying this generic solver without change to different objectives $f_X(\cdot)$, both classic and novel. We require only the condition that $f_X(\cdot)$ be differentiable in M to enable simple gradient descent methods. However, this choice is a convenience of implementation and not a fundamental issue, and thus approaches for optimization of nondifferentiable objectives over nonconvex sets (here $\mathcal{O}^{d\times r}$) could be readily introduced to remove this restriction (for example, Boyd et al., 2011).

3. Survey of Linear Dimensionality Reduction Methods

We now review linear dimensionality reduction techniques using the framework of Section 2, to understand the problem-specific objective and manifold constraint of each method.

3.1 Linear Dimensionality Reduction with Orthogonal Matrix Constraints

Amongst all dimensionality reduction methods, the most widely used techniques are orthogonal projections. These methods owe their popularity in part due to their simple geometric interpretation as a low-dimensional view of high-dimensional data. This interpretation is of great comfort to many application areas, since these methods do not artificially create or exaggerate many types of structure in the data, as is possible with other models that encode strong prior assumptions.

3.1.1 Principal Component Analysis

Principal component analysis (PCA) was originally formulated by Pearson (1901) as a minimization of the sum of squared residual errors between projected data points and the original data $f_X(M) = ||X - MM^\top X||_F^2 = \sum_{i=1}^n ||x_i - MM^\top x_i||_2^2$. Modern treatments tend to favor the equivalent "maximizing variance" derivation (e.g., Bishop, 2006), resulting in the objective $-\text{tr}(M^\top XX^\top M)$. We write PCA in the formulation of Equation 1 as

$$\begin{array}{ll} \text{minimize} & ||X - MM^\top X||_F^2 \\ \text{subject to} & M \in \mathcal{O}^{d\times r}. \end{array} \tag{2}$$

Equation 2 leads to the familiar SVD solution: after summarizing the data by its sample covariance matrix $\frac{1}{n}XX^\top$, the decomposition $XX^\top = Q\Lambda Q^\top$ produces an optimal point $M = Q_r$, where Q_r denotes the columns of Q associated with the largest r eigenvalues of $XX^\top$ (Eckart and Young, 1936; Mirsky, 1960; Golub and Van Loan, 1996).

There are many noteworthy extensions to PCA. A first example is kernel PCA, which uses PCA on a feature space instead of the inputs themselves (Schölkopf et al., 1999), and indeed some dimensionality reduction methods and their kernelized counterparts can be considered together as kernel regression problems (De la Torre, 2012). While quite important for all machine learning methods, we consider kernelized methods orthogonal to much of the presentation here, since using this kernel mapping is a question of representation of data, not of the dimensionality reduction algorithm itself.

Second, there have been several probabilistic extensions to PCA, such as probabilistic PCA (PPCA; Tipping and Bishop, 1999; Roweis, 1997), extreme component analysis (Welling et al., 2003), and minor component analysis (Williams and Agakov, 2002). These algorithms all share a common purpose (modeling covariance) and the same coordinate system for projection (the principal axes of the covariance ellipsoid), even though they differ in the particulars of the projection and which basis is chosen from that coordinate system. We present PPCA as a separate algorithm below and leave the others as extensions of this core method.

Third, extensions have introduced outlier insensitivity via a different implicit noise model such as a Laplace observation model, leading to a few examples of robust PCA (Galpin and Hawkins, 1987; Baccini et al., 1996; Choulakian, 2006). An alternative approach to robust PCA is driven by the observation that a small number of highly corrupted observations can drastically influence standard PCA. Candes et al. (2011) takes this approach to robust PCA, considering the data as low-rank plus sparse noise. Their results have particular theoretical and practical appeal and connect linear dimensionality reduction to the substantial nuclear-norm minimization literature.

Fourth, PCA has been made sparse in several contexts (Zou et al., 2006; d'Aspremont et al., 2007, 2008; Journee et al., 2010), where the typical PCA objective is augmented with a lasso-type ℓ_1 penalty term, namely $f_X(M) = ||X - MM^\top X||_F^2 + \lambda||M||_1$, with penalty term λ and $||M||_1 = \sum_i \sum_j |M_{ij}|$. This objective does not admit an eigenvalue approach, and as a result several specialized algorithms have been proposed. Note however that this sparse objective is again simply a program over $\mathcal{O}^{d\times r}$ (albeit nondifferentiable).

Fifth, another class of popular extensions generalizes PCA to other exponential family distributions, beyond the implicit normal distribution of standard PCA (Collins et al., 2002; Mohamed et al., 2008). These methods, while important, result in nonlinear mappings of the data and thus fall outside the scope of Definition 1. Additionally, there are other nonlinear extensions to PCA; Chapter 12.6 of Hyvarinen et al. (2001) gives an overview.

3.1.2 Multidimensional Scaling

Multidimensional scaling (MDS; Torgerson, 1952; Cox and Cox, 2001; Borg and Groenen, 2005) is a class of methods and a large literature in its own right, but its connections to linear dimensionality reduction and PCA are so well-known that it warrants individual mention. PCA minimizes low-dimensional reconstruction error, but another sensible objective is to maximize the scatter of the projection, under the rationale that doing so would yield the most informative projection (this choice is sometimes called classical scaling). Defining our projected points $y_i = M^\top x_i$ for some $M \in \mathcal{O}^{d\times r}$, MDS seeks to maximize pairwise distances $\sum_i \sum_j ||y_i - y_j||^2$.

MDS leads to the seemingly novel optimization program (Equation 1) over the scatter objective $f_X(M) = \sum_i \sum_j ||M^\top x_i - M^\top x_j||^2$, which can be expanded as

$$f_X(M) \propto \mathrm{tr}\left(M^\top XX^\top M\right) - 1^\top X^\top MM^\top X1 = \mathrm{tr}\left(M^\top X\left(I - \frac{1}{n}11^\top\right)X^\top M\right), \quad (3)$$

where we denote the vector of all ones as 1. Noting that X has zero mean by definition and thus $X(I - \frac{1}{n}11^\top) = X$, we see classical MDS is precisely the 'maximal variance' PCA objective $\mathrm{tr}(M^\top XX^\top M)$.

The equivalence of MDS and PCA in this special case is well-known (Cox and Cox, 2001; Borg and Groenen, 2005; Mardia et al., 1979; Williams, 2002), and indeed this particular example only scratches the surface of MDS, which is usually considered in much more general terms. Specifically, if we have available only pairwise distances $d_X(x_i, x_j)$, a more general MDS problem statement is to fit the low-dimensional data so as to preserve these pairwise distances as closely as possible in the least squares sense: minimizing $\sum_i \sum_j (d_X(x_i, x_j) - d_Y(y_i, y_j))^2$ is known as Kruskal-Shephard scaling, and the distance metrics can be arbitrary and different between the original and low-dimensional data. First, it is worth noting that least squares is by no means the only appropriate stress function on the distances d_X and d_Y; a Sammon mapping is another common choice (see for example Hastie et al. (2008), §14.8). Second, MDS does not generally require the data itself, but only the pairwise dissimilarities $d_{ij} = d_X(x_i, x_j)$, which is often a useful property. When the data is known, we see here that if we specify a low-dimensional orthogonal projection $Y = M^\top X$, then indeed this objective will result in the class of linear dimensionality reduction programs

$$\begin{aligned} &\text{minimize} \quad \sum_i \sum_j \left(d_X\left(x_i, x_j\right) - d_Y\left(M^\top x_i, M^\top x_j\right)\right)^2 \\ &\text{subject to} \quad M \in \mathcal{O}^{d \times r}. \end{aligned} \tag{4}$$

Special approaches exist to solve this program on a case-by-case basis (Cox and Cox, 2001; Borg and Groenen, 2005). However, by broadly considering Equation 4 as an optimization over orthogonal projections, we again see the motivation for a generic numerical solver for this class of problems, obviating objective-specific methods.

Of course, the low-dimensional data Y need not be a linear mapping of X (indeed, in many cases the original points X are not even available). This more general form of MDS is used in a variety of nonlinear dimensionality reduction techniques, including prominently Isomap (Tenenbaum et al., 2000), as discussed below in Section 3.3.

3.1.3 Linear Discriminant Analysis

Another natural problem-specific objective occurs when the data X has associated class labels, of which Fisher's linear discriminant analysis (LDA; Fisher, 1936; Rao, 1948; modern references include Fukunaga, 1990; Bishop, 2006) is perhaps the most prominent example. The purpose of LDA is to project the data in such a way that separation between classes is maximized. To do so, LDA begins by partitioning the data covariance $XX^\top$ into covariance contributed within each of the c classes (Σ_W) and covariance contributed between the classes (Σ_B) , such that $XX^\top = \Sigma_W + \Sigma_B$ for

$$\Sigma_W = \sum_{i=1}^{n} (x_i - \mu_{c_i})(x_i - \mu_{c_i})^\top \qquad \Sigma_B = \sum_{i=1}^{n} (\mu_{c_i} - \mu)(\mu_{c_i} - \mu)^\top, \tag{5}$$

where μ is the global data mean (here $\mu = 0$ by definition) and μ_{c_i} is the class mean associated with data point x_i. LDA seeks the projection that maximizes between-class variability $\operatorname{tr}\left(M^\top \Sigma_B M\right)$ while minimizing within-class variability $\operatorname{tr}\left(M^\top \Sigma_W M\right)$, leading

to the optimization program

$$\begin{aligned} \text{maximize} \quad & \frac{\text{tr}\left(M^\top \Sigma_B M\right)}{\text{tr}\left(M^\top \Sigma_W M\right)} \\ \text{subject to} \quad & M \in \mathcal{O}^{d\times r}. \end{aligned} \tag{6}$$

This objective appears very much like a generalized Rayleigh quotient, and is so for $r = 1$. In this special case, $M \in \mathcal{O}^{d\times 1}$ can be found as the top eigenvector of $\Sigma_W^{-1}\Sigma_B$, which can be seen by substituting $L = \Sigma_W^{1/2} M$ into Equation 6 above. This one-dimensional LDA projection is appropriate when there are $c = 2$ classes.

A common misconception is that LDA for higher dimensional projections $r > 1$ can be solved with a greedy selection of the top r eigenvectors of $\Sigma_W^{-1}\Sigma_B$. However, this is certainly not the case, as the top r eigenvectors of $\Sigma_W^{-1}\Sigma_B$ will not in general be orthogonal. The eigenvector solution solves the similar but not equivalent objective $\text{tr}\left(\left(M^\top \Sigma_W M\right)^{-1}\left(M^\top \Sigma_B M\right)\right)$ over $M \in \mathbb{R}^{d\times r}$; these two objectives and a few others are nicely discussed in Chapter 10 of Fukunaga (1990). While each of these choices has its merits, in the common case that one seeks a projection of the original data, the orthogonal M produced by solving Equation 6 is more appropriate. Though rarely discussed, this misconception between the trace-of-quotient and the quotient-of-traces has been investigated in the literature (Yan and Tang, 2006; Shen et al., 2007).

The commonality of this misconception adds additional motivation for this work, to survey and consolidate a fragmented literature. Second, this misconception also points out the limitations of eigenvector approaches: even when considered the standard algorithm for a popular method, eigenvalue decompositions may in fact be an inappropriate choice. Third, as Equation 6 is a simple program over orthogonal projections, we see again the utility of a generic solver, an approach which should outperform traditional approaches (and indeed does, as Section 4 will show).

In terms of extensions, we note a few key constraints of classical LDA: each data point must be labeled with a class (no missing observations), each data point must be labeled with only one class (no mixed membership), and the class boundaries are modeled as linear. As a first extension, one might have incomplete class labels; Yu et al. (2006) extends LDA (with a probabilistic PCA framework; see Section 3.2.2) to the semi-supervised setting where not all points are labeled. Second, data points may represent a mixture of multiple features, such that one wants to extract a projection where one feature is most discriminable. Brendel et al. (2011) offers a possible solution by marginalizing covariances over each feature of interest. Third, Mika et al. (1999) has extended LDA to the nonlinear domain via kernelization, which has also been well used.

3.1.4 Canonical Correlations Analysis

Canonical correlation analysis (CCA) is a problem of joint dimensionality reduction: given two data sets $X_a \in \mathbb{R}^{d_a\times n}$ and $X_b \in \mathbb{R}^{d_b\times n}$, find low-dimensional mappings $Y_a = P_a X_a$ and $Y_b = P_b X_b$ that maximize the correlation between Y_a and Y_b, namely

$$\rho\left(y_a, y_b\right) = \frac{E\left(y_a^\top y_b\right)}{\sqrt{E\left(y_a^\top y_a\right) E\left(y_b^\top y_b\right)}} = \frac{\text{tr}\left(Y_a Y_b^\top\right)}{\sqrt{\text{tr}\left(Y_a Y_a^\top\right)\text{tr}\left(Y_b Y_b^\top\right)}} = \frac{\text{tr}\left(P_a X_a X_b^\top P_b^\top\right)}{\sqrt{\text{tr}\left(P_a X_a X_a^\top P_a^\top\right)\text{tr}\left(P_b X_b X_b^\top P_b^\top\right)}}. \tag{7}$$

CCA was originally derived in Hotelling (1936); more modern treatments include Muirhead (2005); Timm (2002); Hardoon et al. (2004); Hardoon and Shawe-Taylor (2009). This method in its classical form, which we call *Traditional CCA*, seeks to maximize $\rho(y_a, y_b)$ under the constraint that all variables are uncorrelated and of unit variance: $\frac{1}{n} Y_a Y_a^\top = I$, $\frac{1}{n} Y_b Y_b^\top = I$, and $Y_a Y_b^\top = \Lambda$ for some diagonal matrix Λ. As an optimization program over P_a and P_b, Traditional CCA solves

$$\begin{aligned} \text{maximize} \quad & \frac{\operatorname{tr}\left(P_a X_a X_b^\top P_b^\top\right)}{\sqrt{\operatorname{tr}\left(P_a X_a X_a^\top P_a^\top\right) \operatorname{tr}\left(P_b X_b X_b^\top P_b^\top\right)}} \\ \text{subject to} \quad & \frac{1}{n} P_a X_a X_a^\top P_a^\top = I \\ & \frac{1}{n} P_b X_b X_b^\top P_b^\top = I \\ & P_a X_a X_b^\top P_b^\top = \Lambda. \end{aligned} \tag{8}$$

Using the substitution $P_a = M_a^\top \left(X_a X_a^\top\right)^{-1/2}$ for $M_a \in \mathcal{O}^{d_a \times r}$ (and similar for P_b), Traditional CCA reduces to the well known objective

$$\begin{aligned} \text{maximize} \quad & \operatorname{tr}\left(M_a^\top (X_a X_a^\top)^{-1/2} X_a X_b^\top (X_b X_b^\top)^{-1/2} M_b\right) \\ \text{subject to} \quad & M_a \in \mathcal{O}^{d_a \times r} \\ & M_b \in \mathcal{O}^{d_b \times r}. \end{aligned} \tag{9}$$

This objective is maximized when $M_a^\top$ is the top r left singular vectors and $M_b^\top$ is the top r right singular vectors of $(X_a X_a^\top)^{-1/2} X_a X_b^\top (X_b X_b^\top)^{-1/2}$. The linear transformations optimizing Equation 8 are then calculated as $P_a = M_a^\top (X_a X_a^\top)^{-1/2}$, and similar for P_b. This solution is provably optimal for any dimensionality r under the imposed constraints (Muirhead, 2005).

It is apparent by construction that P_a and P_b do not in general represent orthogonal projections (except when $X_a X_a^\top = I$ and $X_b X_b^\top = I$, respectively), and thus Traditional CCA is unsuitable for common settings (such as visualization of data in an orthogonal axis) where an orthogonal mapping is required. In these cases, a common heuristic approach is to orthogonalize P_a and P_b to produce orthogonal mappings of the data $Y_a = M_a^\top X_a$ and $Y_b = M_b^\top X_b$. This heuristic choice, however, produces suboptimal results for the original correlation objective of Equation 7 for all dimensions $r > 1$ (the $r = 1$ case is trivially an orthogonal projection), as the results will show.

Our approach addresses a desire for orthogonal projections directly: with the optimization framework of Equation 1, we can immediately write down a novel linear dimensionality reduction method that preserves Hotelling's original objective but is properly generalized to produce orthogonal projections. We call this method *Orthogonal CCA*, maximizing the correlation $\rho(y_a, y_b)$ objective directly over orthogonal matrices, namely

$$\begin{aligned} \text{maximize} \quad & \frac{\operatorname{tr}\left(M_a^\top X_a X_b^\top M_b\right)}{\sqrt{\operatorname{tr}\left(M_a^\top X_a X_a^\top M_a\right) \operatorname{tr}\left(M_b^\top X_b X_b^\top M_b\right)}} \\ \text{subject to} \quad & M_a \in \mathcal{O}^{d_a \times r} \\ & M_b \in \mathcal{O}^{d_b \times r}. \end{aligned} \tag{10}$$

The resulting low-dimensional mappings are then the orthogonal projections that we desire: $Y_a = M_a^\top X_a$ and $Y_b = M_b^\top X_b$. The optimization program of Equation 10 can not be solved with a known matrix decomposition, thus requiring a direct optimization approach. More importantly, we point out the meaningful difference between Traditional CCA and Orthogonal CCA: Traditional CCA whitens each data set X_a and X_b, and then orthogonally projects these whitened data into a common space such that correlation is maximized. Orthogonal CCA on the other hand preserves the covariance of the original data X_a and X_b, finding orthogonal projections where correlation is maximized without the initial whitening step. It is unsurprising then that these two methods should return different mappings, even when the Traditional CCA result is orthogonalized post hoc. Accordingly, CCA demonstrates the utility of considering linear dimensionality reduction in the framework of Equation 1; methods can be directly written down for the objective and projection of interest, without having to shoehorn the problem into an eigenvector decomposition.

3.1.5 Maximum Autocorrelation Factors

There are a number of linear dimensionality reduction methods that seek to preserve temporally interesting structure in the projected data. A first simple example is maximum autocorrelation factors (MAF; Switzer and Green, 1984; Larsen, 2002). Suppose the high-dimensional data $X \in \mathbb{R}^{d\times n}$ has data points x_t for $t \in \{1, ..., n\}$, and that the index label t defines an order in the data. In such a setting, the structure of interest for the low-dimensional representation may have nothing to do with modeling data covariance (like PCA), but rather the appropriate description should include temporal structure.

Assume that there is an underlying r-dimensional temporal signal that is smooth, and that the remaining $d-r$ dimensions are noise with little temporal correlation (less smooth). MAF then seeks an orthogonal projection $P = M^\top$ for $M \in \mathcal{O}^{d\times r}$ so as to maximize correlation between adjacent points $y_t, y_{t+\delta}$, yielding the objective

$$f_X(M) = \rho(y_t, y_{t+\delta}) = \frac{E(y_t^\top y_{t+\delta})}{\sqrt{E(y_t^2)E(y_{t+\delta}^2)}} = \frac{E(x_t^\top MM^\top x_{t+\delta})}{E(x_t^\top MM^\top x_t)} = \frac{\mathrm{tr}(M^\top \Sigma_\delta M)}{\mathrm{tr}(M^\top \Sigma M)}, \tag{11}$$

where Σ is the empirical covariance of the data $E(x_t x_t^\top) = \frac{1}{n} XX^\top$ and Σ_δ is the symmetrized empirical cross-covariance of the data evaluated at a one-step time lag $\Sigma_\delta = \frac{1}{2}\left(E(x_{t+\delta}x_t^\top) + E(x_t x_{t+\delta}^\top)\right)$. This objective results in the linear dimensionality program

$$\begin{aligned} \text{maximize} \quad & \frac{\mathrm{tr}(M^\top \Sigma_\delta M)}{\mathrm{tr}(M^\top \Sigma M)} \\ \text{subject to} \quad & M \in \mathcal{O}^{d\times r}. \end{aligned} \tag{12}$$

Note again the appearance of the quotient-of-traces objective (as in LDA and CCA). Indeed, the same heuristic (solving the trace-of-quotient problem) is typically applied to MAF, which results in the standard choice of the top eigenvectors of $\Sigma^{-1}\Sigma_\delta$ as the solution to Equation 12. Though correct in the $r = 1$ case, this misconception is incorrect for precisely the same reasons as above with LDA, and its use results in the same pitfalls. Directly solving the manifold optimization of Equation 1 presents a more straightforward option.

MAF can be seen as a method balancing the desire for cross-covariance (Σ_δ) of the data without overcounting data that has high power (the denominator containing Σ). Indeed, such methods have been invented with slight variations in various application areas (e.g., Cunningham and Yu, 2014). For example, one might simply ask to maximize the cross-covariance $E(y_t^\top y_{t+\delta})$ rather than the correlation itself. Doing so results in a simpler problem than Equation 12: maximize $\text{tr}(M^\top \Sigma_\delta M)$ for $M \in \mathcal{O}^{d\times r}$. In this case the eigenvector solution is optimal. Second, we may want to maximize (or minimize, as in Turner and Sahani (2007)) the squared distance between projected points; the objective then becomes $E(||y_{t+\delta} - y_t||^2)$, which through a similar set of steps produces the similar eigenvalue problem $\text{tr}\left(M^\top(\Sigma - \Sigma_\delta)M\right)$ for $M \in \mathcal{O}^{d\times r}$. This last choice is a discrete time analog of a more popular method—slow feature analysis—which we discuss in the next section. Third, one might want to specify a particular form of temporal structure in terms of a dynamics objective $f_X(M)$, and seek linear projections containing that structure. The advantage of such an approach is that one can specify a range of dynamical structures well beyond the statistics captured by an autocorrelation matrix. A recent simple example is Churchland et al. (2012), who sought a linear subspace of the data where linear dynamics were preserved, namely an M minimizing $f_X(M) = ||\dot{X} - MDM^\top X||_F^2$ for some dynamics matrix $D \in \mathbb{R}^{r\times r}$. This objective is but one simple choice of dynamical structure; given the canonical autonomous system $\dot{y} = g(y)+\epsilon$, one might similarly optimize $f_X(M) = ||M^\top \dot{X} - g(M^\top X)||_F^2$. Optimizing such a program finds the projection of the data that optimally expresses that dynamical feature of interest, without danger of artificially creating that structure based on a strong prior model (as is possible in state-space models like the Kalman filter).

3.1.6 Slow Feature Analysis

Similar in spirit to MAF, slow feature analysis (SFA; Wiskott and Sejnowski, 2002; Wiskott, 2003) is a linear dimensionality reduction technique developed to seek invariant representations in object recognition problems. SFA assumes that measured data, such as pixels in a movie, can have rapidly changing values over time, whereas the identity, pose, or position of the underlying object should move much more slowly. Thus, recovering a slowly moving projection of data may produce a meaningful representation of the true object of interest. Accordingly, assuming access to derivatives $\dot{X} = [\dot{x}_1, ..., \dot{x}_n]$, SFA minimizes the trace of the covariance of the projection $\text{tr}(\dot{Y}\dot{Y}^\top) = \text{tr}(M^\top \dot{X}\dot{X}^\top M)$. This objective is PCA on the derivative data:

$$\begin{aligned} &\text{minimize} && \text{tr}\left(M^\top \dot{X}\dot{X}^\top M\right) \\ &\text{subject to} && M \in \mathcal{O}^{d\times r}. \end{aligned} \tag{13}$$

Linear SFA is the most straightforward case of the class of SFA methods. Several additional choices are typical in SFA implementations, including: *(i)* data points $x_t \in X$ are usually expanded nonlinearly via some feature mapping $h : \mathbb{R}^d \to \mathbb{R}^p$ for some $p > d$ (a typical choice is all monomials of degree one and two to capture linear and quadratic effects); and *(ii)* data are whitened to prevent the creation of structure due to the mapping $h(\cdot)$ alone before the application of the PCA-like program in Equation 13. A logical extension of this nonlinear feature space mapping is to consider a reproducing kernel Hilbert space mapping, as has indeed been done (Bray and Martinez, 2002). Turner and Sahani (2007) established the connections between SFA and linear dynamical systems, giving a probabilistic interpre-

tation of SFA that also makes different and interesting connections of this method to PCA and its probabilistic counterpart (Section 3.2.2).

3.1.7 Sufficient Dimensionality Reduction

Consider a supervised learning problem with high dimensional covariates $X \in \mathbb{R}^{d \times n}$ and responses $Z \in \mathbb{R}^{\ell \times n}$. The concept behind sufficient dimensionality reduction is to find an orthogonal projection of the data $Y = M^\top X \in \mathbb{R}^{r \times n}$ such that the reduced-dimension points Y capture all statistical dependency between X and Z. Thus, sufficient dimensionality reduction (SDR) is a problem of feature selection that seeks an $M \in \mathcal{O}^{d \times r}$ which makes covariates and responses conditionally independent:

$$p_{Z|X}(z|x) = p_{Z|M^\top X}(z|M^\top x) \quad \Longleftrightarrow \quad Z \perp\!\!\!\perp X | M^\top X. \tag{14}$$

SDR is in fact a class of methods, as there are a number of ways one might derive an objective for such a conditional independence relationship. Particularly popular in machine learning is the use of kernel mappings to characterize the conditional independence relationship of Equation 14 (Fukumizu et al., 2004, 2009; Nilsson et al., 2007). The essential idea in these works is to map covariates X and responses Z into reproducing kernel Hilbert spaces, where it has been shown that, for universal kernels, cross-covariance operators can be used to determine conditional independence of X and Z (Fukumizu et al., 2004; Gretton et al., 2012, 2005). Such an approach induces the cost function on the projection

$$f_X(M) = J(Z, M^\top X) := \operatorname{tr}\left(\bar{K}_Z \left(\bar{K}_{M^\top X} + n\epsilon I\right)^{-1}\right), \tag{15}$$

where $\bar{K}_Z = \left(I - \frac{1}{n} 1 1^\top\right) K_Z \left(I - \frac{1}{n} 1 1^\top\right)$ is the centered Gram matrix $K_Z = \{k(z_i, z_j)\}_{ij}$ (and similar for $\bar{K}_{M^\top X}$). Critically, this cost function is provably larger than $J(Z, X)$, with equality if and only if the desired conditional independence of Equation 14 holds. Thus, we have the following linear dimensionality reduction program:

$$\begin{aligned} &\text{minimize} && \operatorname{tr}\left(\bar{K}_Z \left(\bar{K}_{M^\top X} + n\epsilon I\right)^{-1}\right) \\ &\text{subject to} && M \in \mathcal{O}^{d \times r}. \end{aligned} \tag{16}$$

SDR has been extended to the unsupervised case (Wang et al., 2010) and has been implemented with other objectives such as the Hilbert-Schmidt independence criterion (Gretton et al., 2005). An important review of non-kernel SDR techniques is Adragni and Cook (2009), in addition to earlier work (Li, 1991).

3.1.8 Locality Preserving Projections

All methods considered thus far stipulate objectives based on global loss functions, which can be sensitive to outliers and can be significantly distorted by nonlinear structure in the data. A popular alternative throughout machine learning is to consider local neighborhood structure. In the case of dimensionality reduction, considering locality often amounts to constructing a neighborhood graph of the training data, and using that graph to define the loss function. Numerous *nonlinear* methods have been proposed along these lines (see Section 3.3), and this development has led to a few important *linear* methods that consider

local structure. First, locality preserving projections (LPP) (He and Niyogi, 2004) is a direct linear interpretation of Laplacian Eigenmaps (Belkin and Niyogi, 2003). LPP begins by defining a graph with each data point $x_i \in \mathbb{R}^d$ as a vertex, connecting x_i and x_j with the edge $\delta_{i,j}$ if these points are in the same ϵ neighborhood (that is, $||x_i - x_j|| < \epsilon$). A kernel (typically the squared exponential kernel) is then used to weight the existing edges. The cost of the reconstruction $y_i = Px_i$ is then

$$\sum_{i=1}^{n}\sum_{j=1}^{n}||Px_i - Px_j||_2^2 W_{ij} \quad , \quad \text{where} \quad W_{ij} = \delta_{i,j}\exp\left\{-\frac{1}{\tau}||x_i - x_j||_2^2\right\}. \tag{17}$$

Through a few standard steps (see for example Belkin and Niyogi, 2003; He and Niyogi, 2004), this objective results in the linear dimensionality objective

$$\begin{aligned} &\text{minimize} && \text{tr}\left(PXLX^\top P^\top\right) \\ &\text{subject to} && PXDX^\top P^\top = I, \end{aligned} \tag{18}$$

where the matrix D is diagonal with the column sums of W, namely $D_{ii} = \sum_j W_{ij}$, and $L = D - W$ is the Laplacian matrix. Note that this constraint set is sometimes called a *flag* matrix manifold. As in Traditional CCA (Section 3.1.4), LPP can be solved to produce the matrix P with columns equal to the generalized eigenvectors v_i satisfying $XLX^\top v_i = \lambda_i XDX^\top v_i$, by implicitly solving an orthogonally constrained optimization over $M = (XDX^\top)^{1/2}P^\top \in \mathcal{O}^{d\times r}$:

$$\begin{aligned} &\text{minimize} && \text{tr}\left(M^\top (XDX^\top)^{-\top/2} XLX^\top (XDX^\top)^{-1/2} M\right) \\ &\text{subject to} && M \in \mathcal{O}^{d\times r}. \end{aligned} \tag{19}$$

Note again that the resulting linear mapping $Y = PX = M^\top (XDX^\top)^{-\top/2} X$ is not an orthogonal projection.

Related to LPP, neighborhood preserving embedding (NPE) (He et al., 2005) has a largely parallel motivation. NPE is a linear analogue to locally linear embedding (Roweis and Saul, 2000) that produces a different linear approximation to the Laplace Beltrami operator, resulting in the objective

$$\begin{aligned} &\text{minimize} && \text{tr}\left(M^\top (XX^\top)^{-\top/2} X(I-W)^\top (I-W) X^\top (XX^\top)^{-1/2} M\right) \\ &\text{subject to} && M \in \mathcal{O}^{d\times r}, \end{aligned} \tag{20}$$

where the matrix W is the same in LPP. We group these methods together due to their similarity in motivation and resulting objective.

3.2 Linear Dimensionality Reduction with Unconstrained Objectives

All methods reviewed so far involve orthogonal mappings, but several methods simplify further to an unconstrained optimization over matrices $M \in \mathbb{R}^{d\times r}$. We describe those linear dimensionality reduction methods here.

3.2.1 Undercomplete Independent Component Analysis

Independent Component Analysis (ICA; Hyvarinen et al., 2001) is a massively popular class of methods that is often considered alongside PCA and other simple linear transformations. ICA specifies the usual data $X \in I\!R^{d \times n}$ as a mixture of unknown and independent sources $Y \in I\!R^{r \times n}$. Note the critical difference between the independence requirement and the uncorrelatedness of PCA and other methods: for each source data point $y = [y^1, ..., y^r]^\top \in I\!R^r$ (one column of Y), independence implies $p(y) \approx \prod_{j=1}^{r} p\left(y^j\right)$, where the $p\left(y^j\right)$ are the univariate marginals of the low dimensional data (sources).

ICA finds the demixing matrix P such that we recover the independent sources as $Y = PX$. The vast majority of implementations and presentations of ICA deal with the dimension preserving case of $r = d$, and indeed most widely used algorithms require this parity. In this case, ICA is not a dimensionality reduction method.

Our case of interest for dimensionality reduction is the 'undercomplete' case where $r < d$, in which case $Y = PX$ is a linear dimensionality reduction method according to Definition 1. Interestingly, the most common approach to undercomplete ICA is to preprocess the mixed data X with PCA (e.g., Joho et al., 2000), reducing the data to r dimensions, and running a standard square ICA algorithm. That said, there are a number of principled approaches to undercomplete ICA, including (Stone and Porrill, 1998; Zhang et al., 1999; Amari, 1999; De Ridder et al., 2002; Welling et al., 2004). All of these models necessarily involve a probabilistic model, required by the independence of the sources. As an implementation detail, note that observations X are whitened as a preprocessing step.

With this model, authors have maximized the log-likelihood of a generative model (De Ridder et al., 2002) or minimized the mutual information between the sources (Stone and Porrill, 1998; Amari, 1999; Zhang et al., 1999), each of which requires an approximation technique. Welling et al. (2004) describes an exact algorithm for maximizing the log-likelihood of a product of experts objective

$$f_X(M) \quad = \quad \frac{1}{n}\sum_{i=1}^{n} \log p(x_i) \quad \propto \quad \frac{1}{2}\log|M^\top M| + \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{r} \log p_\theta\left(m_k^\top x_i\right), \qquad (21)$$

where m_k are the (unconstrained) columns of M, and $p_\theta(\cdot)$ is a likelihood distribution (an "expert") parameterized by some θ_k. Thus this undercomplete ICA, as an optimization program like Equation 1, is a simple unconstrained maximization of $f_X(M)$ over $M \in I\!R^{d \times r}$.

Extensions of ICA are numerous. Insomuch as undercomplete ICA is a special case of ICA, many of these extensions will also be applicable in the undercomplete case; see the reference Hyvarinen et al. (2001).

3.2.2 Probabilistic PCA

One often-noted shortcoming of PCA is that it partitions data into orthogonal signal (the r-dimensional projected subspace) and noise (the $(d-r)$-dimensional nullspace of $M^\top$). Furthermore, PCA lacks an explicit generative model. Probabilistic PCA (PPCA; Tipping and Bishop, 1999; Roweis, 1997; Theobald, 1975) adds a prior to PCA to address both these potential concerns, treating the high-dimensional data to be a linear mapping of the low-dimensional data (plus noise). If we stipulate some latent independent, identically distributed r-dimensional data $y_i \sim \mathcal{N}(0, I_r)$ for $i \in \{1, ..., n\}$, and we presume

that the high-dimensional data is a noisy linear mapping of that low-dimensional data $x_i|y_i \sim \mathcal{N}(My_i, \sigma_\epsilon^2 I)$ for some given or estimated noise parameter σ_ϵ^2. This model yields a natural objective with the total (negative log) data likelihood, namely

$$f_X(M) \;=\; -\log p(X|M) \;\propto\; \log|MM^\top + \sigma_\epsilon^2 I| + \operatorname{trace}\left((MM^\top + \sigma_\epsilon^2 I)^{-1} XX^\top\right). \quad (22)$$

Mapping this onto our dimensionality reduction program, we want to minimize the negative log likelihood $f_X(M)$ over an arbitrary matrix $M \in \mathbb{R}^{d\times r}$. Appendix A of Tipping and Bishop (1999) shows that this objective can be minimized in closed form as $M = U_r(S_r - \sigma_\epsilon^2 I)^{\frac{1}{2}}$ where $\frac{1}{n}XX^\top = USU^\top$ is the singular value decomposition of the empirical covariance, and U_r denotes the first r columns of U (ordered by the singular values). Tipping and Bishop (1999) also show that the noise parameter σ_ϵ^2 can be solved in closed form, resulting in a closed-form maximum likelihood solution to the parameters of PPCA. This closed-form obviates a more conventional expectation-maximization (EM) approach (Dempster et al., 1977), though in practice EM is still used with the Sherman-Morrison-Woodbury matrix inversion lemma for computational advantage when $d \gg r$. Under this statistical model, the low-dimensional mapping of the observed data is the mean of the posterior $p(Y|X)$, which also corresponds to the MAP estimator: $Y = M^\top(MM^\top + \sigma_\epsilon^2 I)^{-1}X$, which again fits the form of linear dimensionality reduction $Y = PX$.

As with PCA, there are a number of noteworthy extensions to PPCA. Ulfarsson and Solo (2008) add an ℓ_2 regularization term to the PPCA objective. This regularization can be viewed as placing a Gaussian shrinkage prior $p(M)$ on the entries of M, though the authors termed this choice more as a penalty term to drive a sparse solution. A different choice of regularization is found in "Directed" PCA (Kao and Van Roy, 2013), where a trace penalty on the inverse covariance matrix is added. Finally, more generally, several of the extensions noted in Section 3.1.1 are also applicable to the probabilistic version.

3.2.3 Factor Analysis

Factor analysis (FA; Spearman, 1904) has become one of the most widely used statistical methods, in particular in psychology and behavioral sciences. FA is a more general case of a PPCA model: the observation noise is fit per observation rather than across all observations, resulting in the following conditional data likelihood: $x_i|y_i \sim \mathcal{N}(My_i, D)$ for a diagonal matrix D, where the matrix M is typically termed factor loadings. This choice can be viewed as a means to add scale invariance to each measurement, at the cost of losing rotational invariance across observations. Following the same steps as in PPCA, we arrive at the linear dimensionality reduction program

$$\text{minimize} \quad \log|MM^\top + D| + \operatorname{trace}\left((MM^\top + D)^{-1} XX^\top\right), \quad (23)$$

which results in a similar linear dimensionality reduction mapping $Y = PX$ for $P = M^\top(MM^\top + D)^{-1}$. Unlike PPCA, FA has no known closed-form solution, and thus an expectation-maximization algorithm (Dempster et al., 1977) or direct gradient method is typically used to find a (local) optimum of the log likelihood. Extensions similar to those for PPCA have been developed for FA (see for example Kao and Van Roy (2013)).

3.2.4 Linear Regression

Linear regression is one of the most basic and popular tools for statistical modeling. Though not typically considered a linear dimensionality reduction method, this technique maps d-dimensional data onto an r-dimensional hyperplane defined by the number of independent variables. Considering d-dimensional data X as being partitioned into inputs and outputs $X = [X_{in}; X_{out}]$ for inputs $X_{in} \in \mathbb{R}^{r \times n}$ and outputs $X_{out} \in \mathbb{R}^{(d-r) \times n}$, linear regression fits $X_{out} \approx M X_{in}$ for some parameters $M \in \mathbb{R}^{(d-r) \times r}$. The standard choice for fitting such a model is to minimize a simple sum-of-squared-errors objective $f_X(M) = ||X_{out} - M X_{in}||_F^2$, which leads to the least squares solution $M = X_{out} X_{in}^\top (X_{in} X_{in}^\top)^{-1}$. In the form of Equation 1, linear regression is

$$\text{minimize} \quad ||X_{out} - M X_{in}||_F^2 \tag{24}$$

This model produces a regressed data set $\hat{X} = [X_{in}; M X_{in}] = [I; M] X_{in}$. Note that $[I; M]$ has rank r (the data lie on a r-dimensional subspace) and thus Definition 1 applies. To find the dimensionality reduction mapping P, we simply take the SVD $[I; M] = U S V^\top$ and set $P = [S V^\top \; 0]$ where 0 is the $(d-r) \times (d-r)$ matrix of zeroes. The low dimensional mapping of the original data X then takes the standard form $Y = PX$. Chapter 3 of Hastie et al. (2008) gives a thorough introduction to linear regression and points out (Equation 3.46) that the least squares solution can be viewed as mapping the output X_{out} in a projected basis. Adragni and Cook (2009) point out linear regression as a dimensionality reduction method in passing while considering the case of sufficient dimensionality reduction (see SDR, Section 3.1.7, for more detail).

An important extension to linear regression is regularization for bias-variance tradeoff, runtime performance, or interpretability of results. The two most popular include adding an ℓ_2 (ridge or Tikhonov regression) or an ℓ_1 penalty (lasso), resulting in the objective

$$\text{minimize} \quad ||X_{out} - M X_{in}||_F^2 + \lambda ||M||_p \tag{25}$$

for some penalty λ. While the ℓ_2 case can be solved in closed form as an augmented least squares, the ℓ_1 case requires a quadratic program (Tibshirani, 1996); though the simple quadratic program formulation scales poorly (Boyd et al., 2011; Bach et al., 2011). Regardless, both methods produce an analogous form as in standard linear regression, resulting in a linear dimensionality reduction $Y = PX$ for $P = [S V^\top \; 0]$ as above.

Another important extension, particularly given the present subject of dimensionality reduction, is principal components regression and partial least squares (Hastie et al., 2008). Principal components regression uses PCA to preprocess the input variables $X_{in} \in \mathbb{R}^{r \times n}$ down to a reduced $\tilde{X}_{in} \in \mathbb{R}^{\tilde{r} \times n}$, where $\tilde{r}$ is chosen by computational constraints, cross-validation, or similar. Standard linear regression is then run on the resulting components. This two-stage method (first PCA, then regression) can produce deeply suboptimal results, a shortcoming which to some extent is answered by partial least squares. Partial least squares is another classical method that trades off covariance of X_{in} (as in the PCA step of principal components regression) and predictive power (as in linear regression). Indeed, partial least squares has been shown to be a compromise between linear regression and principal components regression, using the framework of continuum regression (Stone and Brooks, 1990). Even still, the partial least squares objective is heuristic and is carried

out on r dimensions in a greedy fashion. Bakır et al. (2004) approached the rank-r linear regression problem directly, writing the objective in the form of Equation 1 as

$$\begin{aligned} \text{minimize} \quad & ||X_{out} - M_{out} S M_{in}^\top X_{in}||_F^2 \\ \text{subject to} \quad & M_{out} \in \mathcal{O}^{d_{out} \times r} \\ & M_{in} \in \mathcal{O}^{d_{in} \times r}, \end{aligned} \tag{26}$$

where S is a nonnegative diagonal matrix, and the optimization program is over the variables $\{M_{in}, M_{out}, S\}$. This method can again be solved as an example of Equation 1.

3.2.5 Distance Metric Learning

Distance metric learning (DML) is an important class of machine learning methods that is typically motivated by the desire to improve a classification method. Numerous algorithms—canonical examples include k-nearest neighbors and support vector machines—calculate distances between training points, and the performance of these algorithms can be improved substantially by a judicious choice of distance metric between these points. Many objectives have been proposed to learn these distance metrics; a seminal work is Xing et al. (2002), and thorough surveys of this literature include Kulis (2012); Yang and Jin (2006); Yang (2007).

In the linear case, to generalize beyond Euclidean distance, distance metric learning seeks a Mahalanobis distance $d_M(x_i, x_j) = ||M^\top x_i - M^\top x_j||_2 = ||x_i - x_j||_{MM^\top}$ that improves some objective on training data. When $M \in \mathbb{R}^{d \times d}$ is full rank, this approach is not a dimensionality reduction. However, as is often noted in that literature, a lower rank $M \in \mathbb{R}^{d \times r}$ for $r < d$ implies a linear mapping of the data to some reduced space where classification (or another objective) is hopefully improved, thus implicitly defining a linear dimensionality reduction method.

Numerous methods have been introduced in the DML literature. Here for clarity we survey one representative method in depth and incorporate other popular approaches from this literature thereafter. Large margin nearest neighbors (LMNN; Weinberger et al., 2005; Torresani and Lee, 2006; Weinberger and Saul, 2009) assumes labeled data: (x_i, z_i), such that $z_i \in \{1, ..., C\}$ for the C data classes. LMNN typically begins by identifying a target neighbor set $\eta(i)$ for each data point x_i, which, in the absence of side information, is simply the k nearest neighbors belonging to the same class z_i as point x_i. The key intuition behind LMNN is that a distance metric $d_M(x_i, x_j)$ is desired such that target neighbors are pulled closer together than any points belonging to a different class, ideally with a large margin. Accordingly, LMNN optimizes the objective

$$f_X(M) = \sum_{i=1}^{n} \sum_{j \in \eta(i)} \left(d_M(x_i, x_j)^2 + \lambda \sum_{\ell=1}^{n} \mathbb{1}(z_i \neq z_\ell) \left[1 + d_M(x_i, x_j)^2 - d_M(x_i, x_\ell)^2 \right]_+ \right), \tag{27}$$

where $\mathbb{1}(\cdot)$ is the indicator function for the class labels z_i, z_ℓ, and $[\cdot]_+$ is the hinge loss. Intuitively, the first term of the right hand side pulls target neighbors closer together, while the second term penalizes (with weight λ) any points x_ℓ that are closer to x_i than its target neighbors x_j (plus some margin), and have a different label ($z_i \neq z_\ell$).

As a dimensionality reduction technique, this objective is readily optimized over $M \in \mathbb{R}^{d \times r}$, to produce a low dimensional mapping of the data $Y = M^\top X$. Beyond LMNN, other prominent methods explore slightly different objectives with similar motivations. Examples include relevant component analysis for DML (Bar-Hillel et al., 2003), neighborhood component analysis (Goldberger et al., 2004), collapsing classes (Globerson and Roweis, 2005), discriminative component analysis (Peltonen et al., 2007), latent coincidence analysis (Der and Saul, 2012), and an online, large-scale method (Chechik et al., 2009). Many of these works also offer kerneled extensions for nonlinear DML.

3.3 Scope Limitations

Definition 1 limits our scope and excludes a number of algorithms that could be considered dimensionality reduction methods. Here we consider four prominent cases that fall outside the definition of linear dimensionality reduction.

3.3.1 Nonlinear Manifold Methods

The most obvious methods to exclude from linear dimensionality reduction are nonlinear manifold methods, the most popular of which include Local Linear Embedding (Roweis and Saul, 2000), Isomap (Tenenbaum et al., 2000), Laplacian eigenmaps (Belkin and Niyogi, 2003), maximum variance unfolding (Weinberger and Saul, 2006), t-distributed stochastic neighbor embedding (Van der Maaten and Hinton, 2008), and diffusion maps (Coifman and Lafon, 2006). These methods seek a nonlinear manifold by using local neighborhoods, geodesic distances, or other graph theoretic considerations. Thus, while these methods are an important contribution to dimensionality reduction, they do not produce low-dimensional data as $Y = PX$ for any P. It is worth noting that some of these problems, such as Laplacian eigenmaps, do involve a generalized eigenvector problem in their derivation, though typically those eigenproblems are the direct solution to a stated objective and not the heuristic that is more often seen in the linear setting (and that motivates the use of direct optimization). A concise introduction to nonlinear manifold methods is given in Zhao et al. (2007), an extensive comparative review is Van der Maaten et al. (2009), and a probabilistic perspective on many spectral methods is given in Lawrence (2012).

3.3.2 Nonparametric Methods

One might also consider classical methods from linear systems theory, like Kalman filtering or smoothing (Kalman, 1960), as linear dimensionality reduction methods. Even more generally, nonparametric methods like Gaussian Processes (Rasmussen and Williams, 2006) also bear some similarity. The key distinction with these algorithms is that our definition of linear dimensionality is parametric: $P \in \mathbb{R}^{r \times d}$ is a fixed mapping and does not change across the data set or some other index. Certainly any nonparametric method violates this restriction, as by definition the transformation mapping must grow with the number of data points. In the Kalman filter, for example, the mapping (which is indeed linear) between each point x_i and its low-dimensional projection y_i changes with each data point (based on all previous data), so in fact this method is also a nonparametric mapping that grows with the number of data points n. This same argument applies to most state-space models and subspace identification methods, including the linear quadratic regulator, linear quadratic

Gaussian control, and similar. Hence these other classic methods also fall outside the scope of linear dimensionality reduction.

3.3.3 Matrix Factorization Problems

A few methods discussed in this work have featured matrix factorizations, and indeed there are many other methods that involve such a decomposition in areas like indexing and collaborative filtering. This general class certainly bears similarity to dimensionality reduction, in that it uses a lower dimensional set of factors to reconstruct noisy or missing high-dimensional data (for example, classical latent semantic indexing is entirely equivalent to PCA; Deerwester et al., 1990). A common factorization objective is to find $H \in \mathbb{R}^{d \times r}$ and $Y \in \mathbb{R}^{r \times n}$ such that the product HY reasonably approximates X according to some criteria. The critical difference between these methods and linear dimensionality reduction is that these methods do not in general yield a sensible linear mapping $Y = PX$, but rather the inverse mapping from low-dimension to high-dimension. While this may seem a trivial and invertible distinction, it is not: specifics of the method often imply that the inverse mapping is nonlinear or ill-defined. To demonstrate why this general class of problem falls outside the scope of linear dimensionality reduction, we detail two popular examples: nonnegative matrix factorization and matrix factorization as used in collaborative filtering.

Nonnegative matrix factorization (NMF; Lee and Seung, 1999; sometimes called multinomial PCA; Buntine, 2002), solves the objective $f_X(H, Y) = ||X - HY||$ for a nonnegative linear basis $H \in \mathbb{R}_+^{d \times r}$ and a nonnegative low-dimensional mapping $Y \in \mathbb{R}_+^{r \times n}$. The critical difference with our construction is that NMF is not linear: there is no P such that $Y = PX$ for all points x_i. If we are given H and a test point x_i, we must do the nonlinear solve $y_i = \text{argmin}_{y \geq 0} ||x_i - Hy||_2$. A simple counterexample is to take an existing point x_j and its nonnegative projection y_j (which we assume is not zero). If we then test on $-x_j$, certainly we can not get $-y_j$ as a valid nonnegative projection.

A second example is the broad class of matrix factorization problems as used in collaborative filtering, which includes weighted low-rank approximations (Srebro and Jaakkola, 2003), maximum margin matrix factorization (Srebro et al., 2004; Rennie and Srebro, 2005), probabilistic matrix factorization (Mnih and Salakhutdinov, 2007), and more. As above, collaborative filtering algorithms approximate data X with a low-dimensional factor model HY. However, the goal of collaborative filtering is to fill in the missing entries of X (e.g., to make movie or product recommendations), and indeed the data matrix X is usually missing the vast majority of its entries. Thus, not only is there no explicit dimensionality reduction $Y = PX$, but that operation is not even well defined for missing data.

More broadly, there has been a longstanding literature in linear algebra of low rank approximations and matrix nearness problems, often called Procrustes problems (Higham, 1989; Li and Hu, 2011; Ruhe, 1987; Schonemann, 1966). These optimization programs have the objective $f_X(M) = ||X - M||$ for some norm (often a unitarily invariant norm, most commonly the Frobenius norm) and some constrained, low-rank matrix M. PCA would be an example, considering X as the data (or the covariance) and M as the r-rank approximation thereof. While a few linear dimensionality reduction methods can be written as Procrustes problems, not all can, and thus nothing general can be claimed about the connection between Procrustes problems and the scope of this work.

Method	Objective $f_X(M)$	Manifold $\mathcal{M}$	Mapping $Y = PX$
PCA (§3.1.1)	$\|\|X - MM^\top X\|\|_F^2$	$\mathcal{O}^{d\times r}$	$M^\top X$
MDS (§3.1.2)	$\sum_{i,j}\left(d_X(x_i,x_j) - d_Y(M^\top x_i, M^\top x_j)\right)^2$	$\mathcal{O}^{d\times r}$	$M^\top X$
LDA (§3.1.3)	$\frac{\mathrm{tr}(M^\top \Sigma_B M)}{\mathrm{tr}(M^\top \Sigma_W M)}$	$\mathcal{O}^{d\times r}$	$M^\top X$
Traditional CCA (§3.1.4)	$\mathrm{tr}\left(M_a^\top (X_a X_a^\top)^{-1/2} X_a X_b^\top (X_b X_b^\top)^{-1/2} M_b\right)$	$\mathcal{O}^{d_a\times r} \times \mathcal{O}^{d_b\times r}$	$M_a^\top \left(X_a X_a^\top\right)^{-1/2} X_a$, $M_b^\top \left(X_b X_b^\top\right)^{-1/2} X_b$
Orthogonal CCA (§3.1.4)	$\frac{\mathrm{tr}\left(M_a^\top X_a X_b^\top M_b\right)}{\sqrt{\mathrm{tr}\left(M_a^\top X_a X_a^\top M_a\right)\mathrm{tr}\left(M_b^\top X_b X_b^\top M_b\right)}}$	$\mathcal{O}^{d_a\times r} \times \mathcal{O}^{d_b\times r}$	$M_a^\top X_a$, $M_b^\top X_b$
MAF (§3.1.5)	$\frac{\mathrm{tr}(M^\top \Sigma_\delta M)}{\mathrm{tr}(M^\top \Sigma M)}$	$\mathcal{O}^{d\times r}$	$M^\top X$
SFA (§3.1.6)	$\mathrm{tr}(M^\top \dot{X}\dot{X}^\top M)$	$\mathcal{O}^{d\times r}$	$M^\top X$
SDR (§3.1.7)	$\mathrm{tr}\left(\bar{K}_Z\left(\bar{K}_{M^\top X} + n\epsilon I\right)^{-1}\right)$	$\mathcal{O}^{d\times r}$	$M^\top X$
LPP (§3.1.8)	$\mathrm{tr}\left(M^\top (XDX^\top)^{-\top/2} XLX^\top (XDX^\top)^{-1/2} M\right)$	$\mathcal{O}^{d\times r}$	$M^\top (XDX^\top)^{-\top/2} X$
UICA (§3.2.1)	$\frac{1}{2}\log\|M^\top M\| + \frac{1}{n}\sum_{i=1}^n \sum_{k=1}^r \log f_\theta\left(m_k^\top x_n\right)$	$\mathbb{R}^{d\times r}$	$M^\top X$
PPCA (§3.2.2)	$\log\|MM^\top + \sigma^2 I\| + \mathrm{tr}\left(XX^\top (MM^\top + \sigma^2 I)^{-1}\right)$	$\mathbb{R}^{d\times r}$	$M^\top (MM^\top + \sigma^2 I)^{-1} X$
FA (§3.2.3)	$\log\|MM^\top + D\| + \mathrm{tr}\left(XX^\top (MM^\top + D)^{-1}\right)$	$\mathbb{R}^{d\times r}$	$M^\top (MM^\top + D)^{-1} X$
LR (§3.2.4)	$\|\|X_{out} - MX_{in}\|\|_F^2 + \lambda\|\|M\|\|_p$	$\mathbb{R}^{d\times r}$	$SV^\top X_{in}$ for $M = USV^\top$
DML (§3.2.5)	$\sum_{i,j\in\eta(i)}\Big\{d_M(x_i,x_j)^2 + \lambda\sum_\ell \mathbb{1}(z_i \neq z_\ell)$ $\left[1 + d_M(x_i,x_j)^2 - d_M(x_i,x_\ell)^2\right]_+\Big\}$	$\mathbb{R}^{d\times r}$	$M^\top X$

Table 1: Summary of linear dimensionality reduction methods.

3.4 Summary of the Framework

Table 1 offers a consolidated summary of these methods. Considering linear dimensionality reduction through the lens of a constrained matrix optimization enables a few key insights. First, as is the primary purpose of this paper, this framework surveys and consolidates the space of linear dimensionality reduction methods. It clarifies that linear dimensionality reduction goes well beyond PCA and can require much more than simply eigenvalue decompositions, and also that many of these methods bear significant resemblance to each other in spirit and in detail. Second, this consolidated view suggests that, since optimization programs over well-understood matrix manifolds address a significant subclass of these methods, an objective-agnostic solver over matrix manifolds may provide a useful generic solver for linear dimensionality reduction techniques.

4. Results

All methods considered here have specified $\mathcal{M}$ as either unconstrained matrices or matrices with orthonormal columns, variables in the space $\mathbb{R}^{d\times r}$. In the unconstrained case, numerous standard optimizers can and have been brought to bear to optimize the objective $f_X(M)$. In the orthogonal case, we have also claimed that the very well-understood geometry of the manifold of orthogonal matrices enables optimization over these manifolds. Pursuing such approaches is critical to consolidating and extending dimensionality reduction, as orthogonal projections $Y = M^\top X$ for $M \in \mathcal{O}^{d\times r}$ are arguably the most natural formulation of linear dimensionality reduction: one seeks a low-dimensional view of the data where some feature is optimally preserved.

The matrix family $\mathcal{O}^{d\times r}$ is precisely the real Stiefel manifold, which is a compact, embedded submanifold of $\mathbb{R}^{d\times r}$. In our context, this means that many important intuitions of optimization can be carried over onto the Stiefel manifold. Notably, with a differentiable objective function $f_X(M)$ and its gradient $\nabla_M f$, one can carry out standard first order optimization via a projected gradient method, where the unconstrained gradient is mapped onto the Stiefel manifold for gradient steps and line searches. Second order techniques also exist, with some added complexity. The foundations of these techniques are Luenberger (1972); Gabay (1982), both of which build on classic and straightforward results from differential geometry. More recently, Edelman et al. (1998) sparked significant interest in optimization over matrix manifolds. Some relevant examples include Manton (2002, 2004); Fiori (2005); Nishimori and Akaho (2005); Abrudan et al. (2008); Ulfarsson and Solo (2008); Srivastava and Liu (2005); Rubinshtein and Srivastava (2010); Varshney and Willsky (2011). Indeed, some of these works have been in the machine learning community (Fiori, 2005; Ulfarsson and Solo, 2008; Varshney and Willsky, 2011), and some have made the connection of geometric optimization methods to PCA (Srivastava and Liu, 2005; Ulfarsson and Solo, 2008; Rubinshtein and Srivastava, 2010; Varshney and Willsky, 2011). The basic geometry of this manifold, as well as optimization over Riemannian manifolds, has been often presented and is now fairly standard. For completeness, we include a primer on this topic in Appendix A. There, as a motivating example, we derive the tangent space, the projection operation, and a retraction operation for the Stiefel manifold. Appendix A then includes Algorithm 1, which uses these objects to present an optimization routine that performs gradient descent over the Stiefel manifold. For a thorough treatment, we refer the interested reader to the excellent summary of much of this modern work (Absil et al., 2008).

One important technical note warrants mention here. The Stiefel manifold is the manifold of all ordered r-tuples of orthonormal vectors in $\mathbb{R}^d$, but in some cases the dimensionality reduction objective $f_X(\cdot)$ evaluates only the subspace (orthonormal basis) implied by M, not the particular choice and order of the orthonormal vectors in M. This class of objective functions is precisely those functions $f_X(M)$ such that, for any $r \times r$ orthogonal matrix R, $f_X(M) = f(MR)$. The implied constraint in these cases is the manifold of rank-r subspaces in $\mathbb{R}^d$, which corresponds to the real Grassmann manifold $\mathcal{G}^{d\times r}$ (another very well understood manifold). As a clarifying example, note that the PCA objective is redundant on the Stiefel manifold: if we want the highest variance r-dimensional projection of our data, the parameterization of those r dimensions is arbitrary, and indeed $f(M) = ||X - MM^\top X||_F^2 = f(MR)$ for any orthogonal R. If one is particularly inter-

ested in ranked eigenvectors, there are standard numerical tricks to break this equivalence and produce an ordered result: for example, maximizing $\text{tr}(AM^\top XX^\top M)$ over the Stiefel manifold, where A is any diagonal matrix with ordered elements $(A_{11} > ... > A_{rr})$. From the perspective of optimization and linear dimensionality reduction, the difference between the Grassmann and Stiefel manifold is one of identifiability. Since there is an uncountable set of Stiefel points corresponding to a single Grassmann point, it seems sensible for many reasons to optimize over the Grassmann manifold when possible (though, as our results will show, this distinction empirically mattered very little). Indeed, most of the optimization literature noted above also deals with the Grassmann case, and the techniques are similar. Conveniently, an objective $f_X(M)$ can be quickly tested for the true implied manifold by comparing values of $f_X(MR)$ for various R. Because the end result is still a matrix $M \in \mathcal{O}^{d\times r}$ (which happens to be in a canonical form in the Grassmann case), this fact truly is an implementation detail of the algorithm, not a fundamental distinction between different linear dimensionality reduction methods. Thus, we present our results as agnostic to this choice, and we empirically revisit the question of identifiability at the end of this section.

To demonstrate the effectiveness of these optimization techniques, we implemented a variety of linear dimensionality reduction methods with several solvers: first order steepest descent methods over the Stiefel and Grassmann manifolds, and second order trust region methods over the Stiefel and Grassmann manifolds (Absil et al., 2008). We implemented these methods in MATLAB, both natively for first order methods, and using the excellent `manopt` software library (Boumal et al., 2014) for first and second order methods (all code is available at `http://github.com/cunni/ldr`). All of these solvers accept, as input, data X and any function that evaluates a differential objective $f_X(M)$ and its gradient $\nabla_M f$ at any point $M \in \mathcal{O}^{d\times r}$, and return, as output, an orthogonal M that corresponds to a (local) optimum of the objective $f_X(M)$.

4.1 Example of Eigenvector Suboptimality

We have cautioned throughout the above survey about the suboptimality of heuristic eigenvector solutions. Figure 1 demonstrates this suboptimality for LDA (Section 3.1.3). In each panel (A and B), we simulated data of dimensionality $d = 3$, with $n = 3000$ points, corresponding to 1000 points in each of 3 clusters (shown in black, blue, and red). Data in each cluster were normally distributed with random means (normal with standard deviation 5/2) and random covariance (uniformly distributed orientation and exponentially distributed eccentricity with mean 5). In the left subpanel of panel A, we then calculated the $r = 2$ dimensional projection by orthogonalizing the top two eigenvectors of the matrix $\Sigma_W^{-1}\Sigma_B$ ('Heuristic LDA'). In the right subpanel, we directly optimized the objective of Equation 6 over $\mathcal{O}^{d\times r}$ ('Orthogonal LDA'). We calculate the normalized improvement of the manifold method as

$$-\frac{\left(f_X\left(M^{(orth)}\right) - f_X\left(M^{(eig)}\right)\right)}{\left|f_X\left(M^{(eig)}\right)\right|}. \tag{28}$$

Throughout the results we will call the results of traditional eigenvector approaches $M^{(eig)}$ and the results of our manifold solver $M^{(orth)}$. Figure 1A shows an example where both the heuristic and manifold optimization methods return qualitatively similar results, and indeed

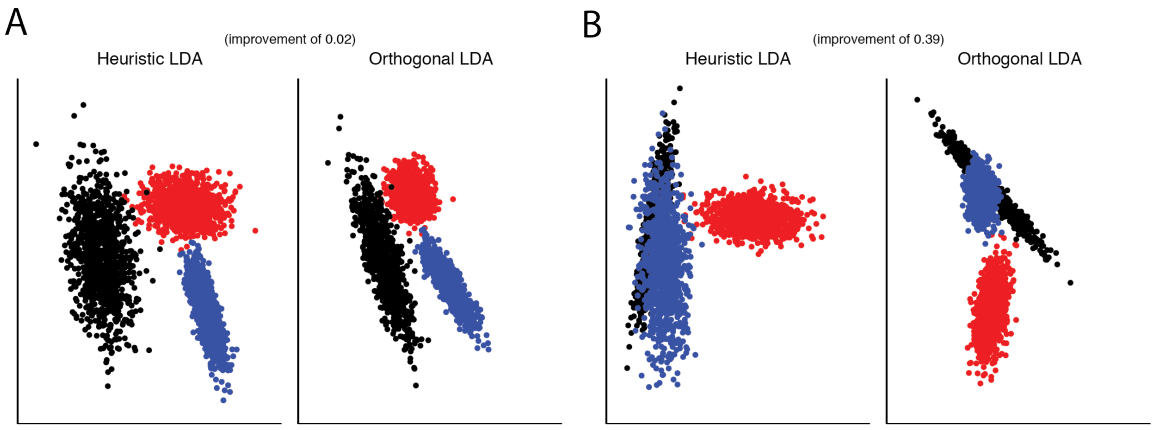


Figure 1: Cautionary example of differences in objectives for LDA. Panel A shows a data set that offers only marginal performance gain by using manifold optimization (Orthogonal LDA, right subpanel of panel A) rather than the traditional eigenvector heuristic (Heuristic LDA, left subpanel). Panel B shows a data set that has a stark difference between the two methods. The measured performance difference (see Equation 28) is shown.

the numerical improvement (0.02) reflects that indeed this heuristic is by no means wildly inappropriate for the stated objective. Indeed, we know it to be correct for $r = 1$. Figure 1B shows a particularly telling example: both methods distinguish the red cluster easily, whereas the heuristic method confounds the black and blue clusters, while the optimization approach offers better separability, which indeed correlates with improvement on the stated objective of Equation 6. It is critical to clarify the distinction between these two methods: the heuristic and orthogonal solutions are indeed optimal, but for *different* objectives, as discussion in Section 3.1.3. Thus, the purpose of this cautionary example is to highlight the importance of optimizing the intended objective, and the freedom to choose that objective without a tacit connection to a generalized eigenvalue problem. These goals can be directly and generically achieved with the optimization framework of Equation 1.

4.2 Performance Improvement

Here we seek to demonstrate the quantitative improvements available by directly optimizing an objective, rather than resorting to an eigenvector heuristic. First we implemented PCA (Section 3.1.1) using both methods. We ran PCA on 20 random data sets for each dimensionality $d \in \{4, 8, 16, ..., 1024\}$, each time projecting onto $r = 3$ dimensions. Data were normally distributed with random covariance (exponentially distributed eccentricity with mean 2). We calculated $f_X\left(M^{(eig)}\right)$ and $f_X\left(M^{(orth)}\right)$ from Equation 2, and we calculated the normalized improvement of the manifold method as above in Equation 28. Since the eigenvector decomposition is provably optimal for PCA, our method should demonstrate no improvement. Indeed, Figure 2 (purple trace) shows the distribution of normalized improvements for PCA is entirely 0 in panel A. We then repeated this analysis for a fixed

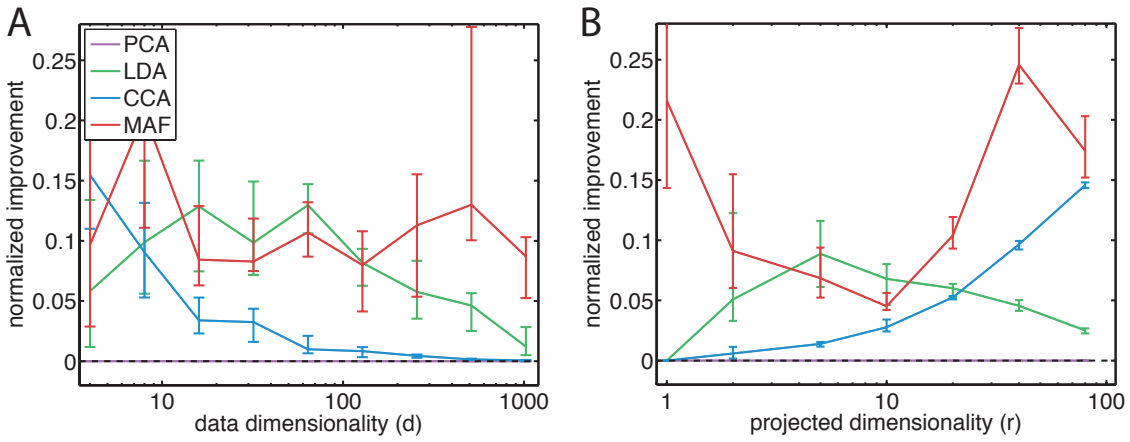


Figure 2: Performance comparison between heuristic solvers and direct optimization of linear dimensionality reduction objectives. The vertical axis denotes normalized improvement of the optimization program over traditional approaches. The error bars show median performance improvement and the central 50th percentile of 20 independent runs at each choice of (d, r).

data dimensionality $d = 100$ (generating data as above), now ranging the projected dimensionality $r \in \{1, 2, 5, 10, 20, 40, 80\}$. These results are shown in Figure 2B, and again, the optimization approach recovers the known PCA optima precisely. This confirmatory result also shows, pleasingly, that there is no empirical downside (in terms of accuracy) to using manifold optimization.

We next repeated the same experiment for LDA (Section 3.1.3). We generated data with 1000 data points in each of d classes, where within class data was generated according to a normal distribution with random covariance (uniformly distributed orientation and exponentially distributed eccentricity with mean 5), and each class mean vector was randomly chosen (normal with standard deviation $5/d$). We compared the suboptimal LDA heuristic $M^{(eig)}$ (orthogonalizing the top r eigenvectors of $\Sigma_W^{-1}\Sigma_B$) to the direct optimization of $f_X(M) = \mathrm{tr}(M^\top \Sigma_B M)/\mathrm{tr}(M^\top \Sigma_W M)$, which produced $M^{(orth)}$. Unlike in PCA, Figure 2 (green traces) shows that directly addressing the LDA objective produces significant performance improvements. The green trace is plotted at the median, and the error bars show the median 50% of the distribution of performance improvements across both data dimensionality d (panel A) and projected dimensionality r (panel B).

We next implemented Traditional CCA and Orthogonal CCA as introduced in Section 3.1.4, which yield the blue performance distributions shown in Figure 2A and B. Data set X_a was generated by a random linear transformation of a latent data set Z (iid standard normal points with dimensionality of $d/2$; the random linear transformation had the same distribution), plus noise, and data set X_b was generated by a different random linear transformation of the same latent Z, plus noise. Again we see significant improvement of direct Orthogonal CCA over orthogonalizing Traditional CCA, when evaluated under the correlation objective of Equation 10. First, we note that to be conservative in this case we omit

the denominator term from the improvement metric (Equation 28); that is, we do not normalize CCA improvements. CCA has a correlation objective, which is already a normalized quantity, and thus renormalizing would increase these improvements. More importantly, it is essential to note that we do not claim any suboptimality of Hotelling's Traditional CCA in solving Equation 8. Rather, it is the subsequent heuristic choice of orthogonalizing the resulting mapping that is problematic. In other words, we show that if one seeks an orthogonal projection of the data, as is often desired in practice, one should do so directly. Our CCA results demonstrate the substantial underperformance of eigenvector heuristics in this case, and our generic solver allows a direct solution without conceptual difficulty.

Finally, we implemented MAF as introduced in Section 3.1.5, where we generated data by a random linear transformation (uniformly distributed entries on $[0, d^{-1/2}]$) of d dimensions of univariate random temporal functions, which we generated with cubic splines with four randomly located knots (uniformly distributed in the domain, standard normally distributed in range), plus noise. MAF is another method that has been solved using an eigenvector heuristic, and the performance improvement is shown in red in Figure 2.

In total, Figure 2 offers some key points of interpretation. First, note that no data lie in the negative halfplane (see black dashed line atop the purple line at 0). Though unsurprising, this is an important confirmation that the optimization program performs unambiguously better than or equal to heuristic methods. Second, methods other than PCA produce approximately 10% improvement using direct optimization, a significant improvement that suggests the broad use of this optimization framework. Third, a natural question for these nonconvex programs is that of local optima. We found that, across a wide range of choices for d and r, nearly all methods converged to the same optimal value whether started at a random M or started at the heuristic point $M^{(eig)}$. Deeper characterization of local optima should be highly dependent on the particular objective and is beyond the scope of this work. Third, we note that methods sometimes have performance equal to the heuristic method; indeed $M^{(eig)}$ is sometimes a local optimum. We found empirically that larger r makes this less likely, and larger d makes this more likely.

A significant point of interpretation is that of size of average performance. We stress that these data sets were not carefully chosen to demonstrate effect. Indeed, we are able to adversarially choose data to create much larger performance improvements, and similarly we can choose data sets that demonstrate no effect. Thus, one should not infer from Figure 2 that, for example, Orthogonal CCA fundamentally has increasing benefit over the heuristic approach with increasing r (or decreasing benefit with increasing d). Instead, we encourage the takeaway of this performance figure to be that one should always optimize the objective of interest directly, rather than resorting to a reasonable but theoretically unsound eigenvector heuristic, as the performance loss is potentially meaningful.

4.3 Computational Cost

Importantly, this matrix manifold solver does not incur massive computational cost. The only additional computation beyond standard unconstrained first-order optimization of dr variables is the projection onto or along the manifold to ensure a feasible $M \in \mathcal{O}^{d \times r}$, which in any scheme requires a matrix decomposition (see Appendix A). Thus each algorithmic step carries an additional cost of $O(dr^2)$. This cost is in many cases dwarfed by the larger

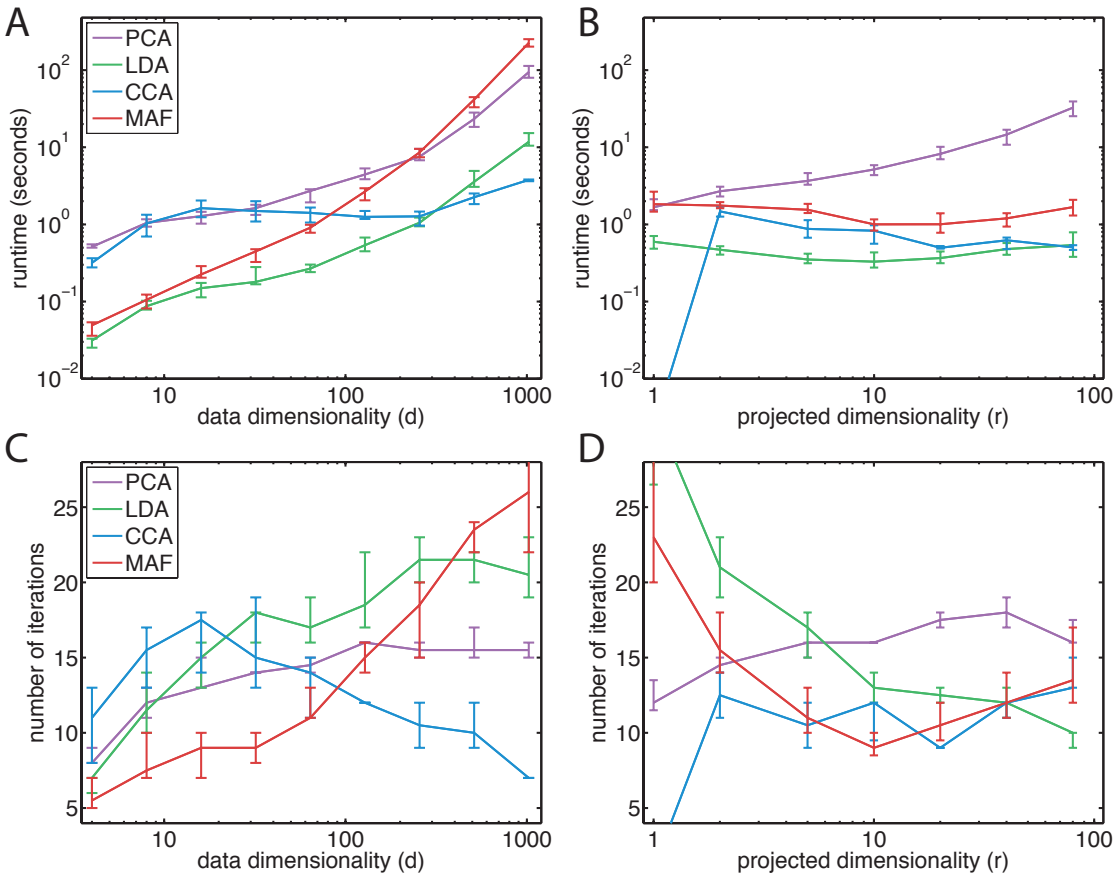


Figure 3: Computational cost of direct optimization of linear dimensionality reduction objectives. Data sets are the same as those in Figure 2. The vertical axis in panels A and B denotes runtime in seconds. Panels C and D show the same data by the number of solver iterations.

cost of calculating matrix-matrix products with a data matrix $X \in \mathbb{R}^{d \times n}$ (which often appear in the gradient calculations $\nabla_M f$). Second order methods approximate or evaluate a Hessian, which incurs more complexity per iteration, but as usual at the tradeoff of drastically fewer iterations. Accordingly, the runtime of manifold optimization is at worst moderately degraded compared to an unconstrained first or second order method. Compared to eigenvector heuristics, which if implemented as a compact SVD cost only $O(dr^2)$, direct optimization is an order of magnitude or more slower due to the iterative nature of the algorithm.

Figure 3 shows the computational cost of these methods, using the same data as in the previous section. In Figure 3A, at each of $d \in \{4, 8, 16, ..., 1024\}$ and for $r = 3$, we ran PCA, LDA, CCA, and MAF 20 times, and we show here the median and central 50% of the runtime distribution (in seconds). This panel demonstrates that runtime increases approximately linearly as expected in d: runtime increases by approximately three orders

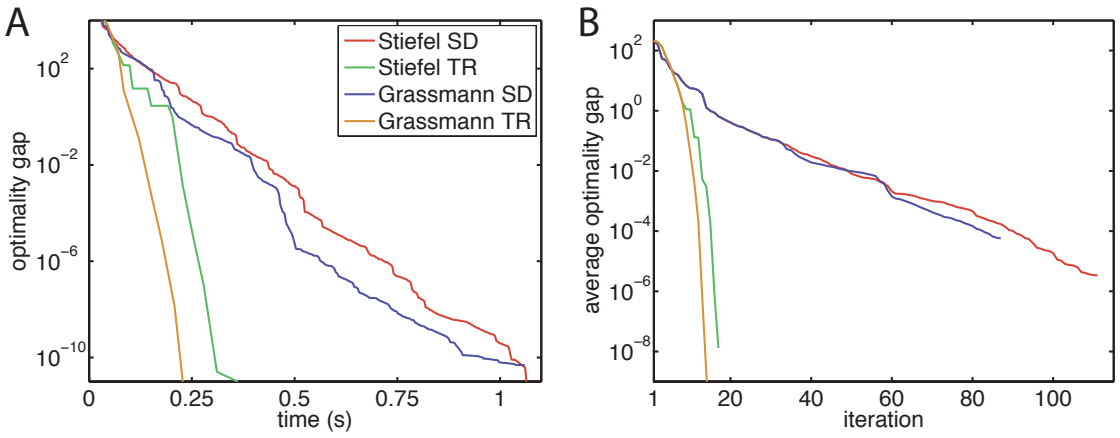


Figure 4: Comparison of different optimization techniques. PCA was run on 100 independent data sets of size $d = 100$, projecting to $r = 10$ dimensions. Panel A shows the median runtime performance (across data sets), with optimality gap as a function of runtime in seconds. Panel B shows the average optimality gap by iteration. PCA was run on each of these data sets independently with the Stiefel steepest descent (red), Stiefel trust region (green), Grassmann steepest descent (blue), and Grassmann trust region (brown) solvers.

of magnitude over three orders of magnitude increase in d. We do a similar simulation in Figure 3B at each of $r \in \{1, 2, 5, 10, 20, 40, 80\}$ for a fixed $d = 100$, and again runtime is increasing.

Figures 3C and 3D show the same data as in Figures 3A and 3B, but by number of solver iterations. In this figure we used a second-order solver over the Grassmann manifold in PCA, LDA, and MAF (critically, the same solver for all three), and the second-order solver over the product of two Stiefel manifolds in the case of CCA. These two panels again underscore the overall point of Figure 3: runtime complexity is not particularly burdensome across a range of reasonable choices for d and r, even with a generic solver.

4.4 Choice of Solver, and Identifiability

We have claimed that the choice of optimization over the Stiefel or Grassmann manifold is a question of identifiability, and further that empirically it seems to matter little to algorithmic performance. Figure 4 gives evidence to that claim. We created 100 independent data sets with $d = 100$ and $r = 10$ for PCA. Here the choice of algorithm is less important, and PCA is a sensible choice because we know the global optimum. We ran PCA using four solvers: first-order steepest descent over the Stiefel manifold, first-order steepest descent over the Grassmann manifold, second-order trust region optimization over the Stiefel manifold, and second-order trust region optimization over the Grassmann manifold. Figure 4 shows the optimality gap by solver choice for each of these four solvers. Figure 4A shows the optimality gap as a function of time for the median performing solver (median across

the 100 independent data sets), and Figure 4B shows the optimality gap (mean across all the 100 independent data sets) as a function of algorithmic iteration. From these figures it is clear that second-order methods outperform first order methods, though perhaps less than one might typically expect. More importantly, the difference between the choice of optimization over the Stiefel or Grassmann manifold is minor at best. This figure, along with previous results, suggest the feasibility of a generic solver for orthogonal linear dimensionality reduction.

5. Discussion

Dimensionality reduction is a cornerstone of data analysis. Among many methods, perhaps none are more often used than the linear class of methods. By considering these methods as optimization programs of user-specified objectives over orthogonal or unconstrained matrix manifolds, we have surveyed a surprisingly fragmented literature, offered insights into the shortcomings of traditional eigenvector heuristics, and have pointed to straightforward generalizations with an objective-agnostic linear dimensionality reduction solver. The results of Section 4 suggest that linear dimensionality reduction can be abstracted away in the same way that unconstrained optimization has been, as a numerical technology that can sometimes be treated as a black-box solver. This survey also suggests that future linear dimensionality reduction algorithms can be derived in a simpler and more principled fashion. Of course, even with such a method one must be careful to design a linear dimensionality reduction sensibly to avoid the many unintuitive pitfalls of high-dimensional data (e.g., Diaconis and Freedman, 1984).

Other authors have surveyed dimensionality reduction algorithms. Some relevant examples include Burges (2010); De la Torre (2012); Sun et al. (2009); Borga et al. (1997). These works all focus on particular subsets of the dimensionality reduction field, and our work here is no different, insomuch as we focus exclusively on linear dimensionality reduction and the connecting concept of optimization over matrix manifolds. Burges (2010) gives an excellent tutorial review of popular methods, including both linear and nonlinear methods, dividing those methods into projective and manifold approaches. De la Torre (2012) surveys five linear and nonlinear methods with their kernelized counterparts using methods from kernel regression. Borga et al. (1997) and Sun et al. (2009) focus on those methods that can be cast as generalized eigenvalue problems, and derive scalable algorithms for those methods, connecting to the broad literature on optimizing Rayleigh quotients.

The simple optimization framework discussed herein offers a direct approach to linear dimensionality reduction: many linear dimensionality reduction methods seek a meaningful, low-dimensional orthogonal subspace of the data, so it is natural to create a program that directly optimizes some objective on the data over these subspaces. This claim is supported by the number of linear dimensionality reduction methods that fit naturally into this framework, by the ease with which new methods can be created, and by the significant performance gains achieved with direct optimization. Thus we believe this survey offers a valuable simplifying principle for linear dimensionality reduction.

This optimization framework is conceptually most similar to the projection index from important literature in projection pursuit (Huber, 1985; Friedman, 1987): both that literature and the present work focus on optimizing objective functions on projections to a lower

dimensional coordinate space. Since the time of the fundamental work in projection pursuit, massive developments in computational power and advances in optimization over matrix manifolds suggest the merit of the present approach. First, the projection pursuit literature is inherently greedy: univariate projections are optimized over the projection index, that structure is removed from the high dimensional data, and the process is repeated. This approach leads to (potentially significant) suboptimality of the results and requires costly computation on the space of the high-dimensional data for structure removal. The present matrix manifold framework circumvents both of these issues. Thus, while the spirit of this framework is very much in line with the idea of a projection index, this framework, both in concept and in implementation, is critically enabled by tools that were unavailable to the original development of projection pursuit.

Acknowledgments

JPC and ZG received funding from the UK Engineering and Physical Sciences Research Council (EPSRC EP/H019472/1). JPC received funding from a Sloan Research Fellowship, the Simons Foundation (SCGB#325171 and SCGB#325233), the Grossman Center at Columbia University, and the Gatsby Charitable Trust.

Appendix A. Optimization over the Stiefel Manifold

Here we offer a basic introduction to optimization over matrix manifolds, restricting our focus to a first-order, projected gradient optimization over the Stiefel manifold $\mathcal{O}^{d\times r}$. Intuitively, manifold projected gradient methods are iterative optimization routines that require firstly an understanding of search directions along the constraint set, called the tangent space (§A.1). With an objective f, gradients $\nabla_M f$ are then calculated in the full space, in this case $\mathbb{R}^{d\times r}$. These gradients are projected onto that tangent space (§A.2). Any nonzero step in a linear tangent space will depart from the nonlinear constraint set, so finally a *retraction* is needed to map a step onto the constraint set (§A.3). With these three components, a standard first-order iterative solver can be carried out, with typical convergence guarantees. We conclude this tutorial appendix with pseudocode in §A.4 and a figure summarizing these steps (Figure 5).

We have previously introduced the Stiefel manifold $\mathcal{O}^{d\times r}$ as the set of all matrices with orthonormal columns, namely $\mathcal{O}^{d\times r} = \left\{M \in \mathbb{R}^{d\times r} : M^\top M = I\right\}$, where I is the $r \times r$ identity matrix. $\mathcal{O}^{d\times r}$ is a manifold, an embedded submanifold of $\mathbb{R}^{d\times r}$, and bounded and closed (and thus compact). From these facts we can carry over all intuitions of an explicit (though nonlinear and nonconvex) constraint set within $\mathbb{R}^{d\times r}$.

A.1 Tangent Space $T_M\mathcal{O}^{d\times r}$

Critical to understanding the geometry of any manifold (in particular to exploit that geometry for optimization) is the *tangent space*, the linear (vector space) approximation to the manifold at a particular point. To define this space, we first define a *curve* on the manifold

$\mathcal{O}^{d\times r}$ as a smooth map $\gamma(\cdot) : \mathbb{R} \to \mathcal{O}^{d\times r}$. Then, the tangent space is

$$T_M\mathcal{O}^{d\times r} = \left\{\dot{\gamma}(0) : \gamma(\cdot) \text{ is a curve on } \mathcal{O}^{d\times r} \text{ with } \gamma(0) = M\right\}, \tag{29}$$

where $\dot{\gamma}$ is the derivative $\frac{d}{dt}\gamma(t)$. Loosely, $T_M\mathcal{O}^{d\times r}$ is the space of directions along the manifold at a point M. While Equation 29 is fairly general for embedded submanifolds, it is abstract and leaves little insight into numerical implementation. Conveniently, the tangent space of the Stiefel manifold has a particularly nice equivalent form.

Claim 1 (Tangent space of the Stiefel Manifold) *The following sets are equivalent:*

$$\begin{aligned} T_M\mathcal{O}^{d\times r} &= \left\{\dot{\gamma}(0) : \gamma(\cdot) \text{ is a curve on } \mathcal{O}^{d\times r} \text{ with } \gamma(0) = M\right\}, && (30)\\ T_1 &= \left\{X \in \mathbb{R}^{d\times r} : M^\top X + X^\top M = 0\right\}, && (31)\\ T_2 &= \left\{MA + (I - MM^\top)B : A = -A^\top, B \in \mathbb{R}^{d\times r}\right\}. && (32) \end{aligned}$$

Proof The proof proceeds in four steps:

1. $X \in T_M\mathcal{O}^{d\times r} \Rightarrow X \in T_1$

   Considering a curve $\gamma(t)$ from Equation 30, we know $\gamma(t)^\top\gamma(t) = I$ (every point of the curve is on the manifold). We differentiate in t to see $\gamma(t)^\top\dot{\gamma}(t) + \dot{\gamma}(t)^\top\gamma(t) = 0$. At $t = 0$, we have $\gamma(0) = M$, and we define the tangent space element $\dot{\gamma}(0) = X$. Then X is such that $M^\top X + X^\top M = 0$.

2. $X \in T_1 \Rightarrow X \in T_M\mathcal{O}^{d\times r}$

   We must construct a curve such that any $X \in T_1$ is a point in the tangent space; consider $\gamma(t) = (M + tX)(I + t^2X^\top X)^{-1/2}$ (a choice that we will see again below in §A.3). First, this curve satisfies $\gamma(0) = M$. Second, $\gamma(\cdot)$ is a curve on the Stiefel manifold, since every point $\gamma(t)$ satisfies

   $$\begin{aligned} \gamma(t)^\top\gamma(t) &= (I + t^2X^\top X)^{-1/2}(M + tX)^\top(M + tX)(I + t^2X^\top X)^{-1/2}\\ &= (I + t^2X^\top X)^{-1/2}(M^\top M + tM^\top X + tX^\top M + t^2X^\top X)(I + t^2X^\top X)^{-1/2}\\ &= (I + t^2X^\top X)^{-1/2}(I + t^2X^\top X)(I + t^2X^\top X)^{-1/2}\\ &= I, \end{aligned}$$

   where the third line uses $M \in \mathcal{O}^{d\times r}$ and $X \in T_1$. It remains to show only that $\dot{\gamma}(0) = X$. We differentiate $\gamma(t)$ as

   $$\dot{\gamma}(t) = X(I + t^2X^\top X)^{-1/2} + (M + tX)\frac{d}{dt}(I + t^2X^\top X)^{-1/2}. \tag{33}$$

   The rightmost derivative term of Equation 33 does not have a closed form, but is the unique solution to a Sylvester equation. Letting $\alpha(t) = (I + t^2X^\top X)^{-1/2}$, we seek

$\dot{\alpha}(0)$. By implicit differentiation,

$$\begin{aligned}\left[\frac{d}{dt}\alpha(t)\alpha(t)\right]_{t=0} &= \left[\frac{d}{dt}(I+t^2X^\top X)^{-1}\right]_{t=0}\\ \dot{\alpha}(0)\alpha(0)+\alpha(0)\dot{\alpha}(0) &= \left[(I+t^2X^\top X)^{-1}\left(2tX^\top X\right)(I+t^2X^\top X)^{-1}\right]_{t=0}\\ 2\dot{\alpha}(0) &= 0,\end{aligned}$$

since $\alpha(0)=I$. Thus we see $\dot{\alpha}(0)=\left[\frac{d}{dt}(I+t^2X^\top X)^{-1/2}\right]_{t=0}=0$. Equation 33 yields $\dot{\gamma}(0)=X$, which completes the proof of the converse.

3. $X\in T_2 \Rightarrow X\in T_1$

   Let $X=MA+(I-MM^\top)B$ according to Equation 32. Then

$$\begin{aligned}M^\top X+X^\top M &= M^\top MA+M^\top(I-MM^\top)B+A^\top M^\top M+B^\top(I-MM^\top)M\\ &= A+A^\top\\ &= 0,\end{aligned}$$

   by the skew-symmetry of A and $M\in\mathcal{O}^{d\times r}$.

4. $X\in T_1 \Rightarrow X\in T_2$

   We show the transposition $X\notin T_2 \Rightarrow X\notin T_1$. By the definition of T_2, $X=MA+(I-MM^\top)B$ is not in T_2 if and only if $A\neq -A^\top$. Then, using the previous argument, we see that such an X has $M^\top X+X^\top M\neq 0$, and thus is not a member of T_1.

Thus, the three tangent space definitions Equations 30-32 are equivalent. The definition of Equation 32 is particularly useful as it is constructive, which is essential when considering optimization. ■

A.2 Projection $\pi_M : \mathbb{R}^{d\times r}\to T_M\mathcal{O}^{d\times r}$

Because $\mathcal{O}^{d\times r}$ is an embedded submanifold of $\mathbb{R}^{d\times r}$, it is natural to consider the metric implied by Euclidean space : $\mathbb{R}^{d\times r}$ endowed with the standard inner product $\langle P,N\rangle = \mathrm{tr}(P^\top N)$, and the induced Frobenius norm $||\cdot||_F$. With this metric, the Stiefel manifold is then a Riemannian submanifold of Euclidean space. This immediately allows us to consider the projection of an arbitrary vector $Z\in\mathbb{R}^{d\times r}$ onto the tangent space $T_M\mathcal{O}^{d\times r}$, namely

$$\begin{aligned}\pi(Z) &= \operatorname*{arg\,min}_{X\in T_M\mathcal{O}^{d\times r}} ||Z-X||_F\\ &= \arg\min ||Z-(MA-(I-MM^\top)B)||_F\\ &= \arg\min ||(MM^\top Z-MA)+(I-MM^\top)(Z-B)||_F\\ &= \arg\min ||M(M^\top Z-A)||_F+||(I-MM^\top)(Z-B)||_F\\ &= \arg\min ||M^\top Z-A||_F+||(I-MM^\top)(Z-B)||_F,\end{aligned}$$

where the last equality comes from the unitary invariance of the Frobenius norm. This expression is minimized by setting $B = Z$ and setting A to be the skew-symmetric part of $M^\top Z$, namely $A := \text{skew}(M^\top Z) = \frac{1}{2}(M^\top Z - Z^\top M)$ (Fan and Hoffman, 1955). This results in the projection

$$\pi_M(Z) = M\text{skew}(M^\top Z) + (I - MM^\top)Z. \tag{34}$$

We note that an alternative canonical metric is often considered in this literature, namely $\langle P, N\rangle_M = \text{tr}\left(P^\top(I - MM^\top)N\right)$. The literature is divided on this choice; for simplicity we choose the standard inner product.

A.3 Retraction $r_M : T_M\mathcal{O}^{d\times r} \to \mathcal{O}^{d\times r}$

Projected gradient methods seek an iterative step in the direction of steepest descent along the manifold, namely $M + \beta\pi_M(-\nabla_M f)$. For any nonzero step size β, this iterate will leave the Stiefel manifold. Thus, a *retraction* is required to map onto the manifold. A number of projective retractions are available (Kaneko et al., 2013); here we define the retraction of a step Z away from a current manifold point M as

$$r_M(Z) = \underset{N\in\mathcal{O}^{d\times r}}{\arg\min} ||N - (M + Z)||_F, \tag{35}$$

that is, the closest point on the manifold to the desired iterate $M + Z$. For unitarily invariant norms, a classic result is that $r_M(Z) = UV^\top$, where $(M + Z) = USV^\top$ is the singular value decomposition (Fan and Hoffman, 1955), or equivalently, $r_M(Z) = W$ for a polar decomposition $(M + Z) = WP$. Conveniently, when $Z \in T_M\mathcal{O}^{d\times r}$, this retraction has the simple closed form $r_M(Z) = (M + Z)(I + Z^\top Z)^{-1/2}$ (Kaneko et al., 2013), which explains the choice of curve in §A.1.

In the cases of the Stiefel and Grassmann manifolds, it is possible to directly calculate a manifold geodesic (shortest path between two points in the manifold). While more aesthetically pleasing, calculating such a geodesic requires a matrix exponential, and thus has similar computational burden as a projective retraction (often the exponential is slightly more expensive). Empirically, we have found very little difference in the convergence or computational burden of this choice, and thus we focus this tutorial on the conceptually simpler retraction. Absil and Malick (2012) discuss projective retractions compared with geodesics/exponential maps.

A.4 Pseudocode for a Projected Gradient Solver

Algorithm 1 gives pseudocode for a projected gradient method over the Stiefel manifold. This generic algorithm requires only a choice of convergence parameters and a line search method, choices which are standard for first-order optimization. Chapter 4 of Absil et al. (2008) offers a global convergence proof for such a method using Armijo line search. Indeed, the only particular consideration for this algorithmic implementation is the tangent space $T_M\mathcal{O}^{d\times r}$, the projection π_M, and the retraction r_M.

It is worth noting that the above operations imply a two-stage gradient step: Algorithm 1 first projects the free gradient onto the tangent space (π_M), and second the proposed step is retracted onto the manifold (r_M). It is natural to ask why one does not perform

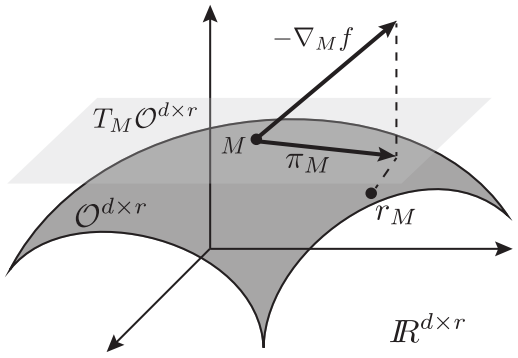


Figure 5: Cartoon of a projected gradient step on the Stiefel manifold. Notation follows Algorithm 1.

Algorithm 1 Gradient descent over the Stiefel manifold (with line search and retraction)

initialize $M \in \mathcal{O}^{d\times r}$
while $f(M)$ has not converged **do**
  calculate $\nabla_M f \in \mathbb{R}^{d\times r}$ # free gradient of objective
  calculate $\pi_M(-\nabla_M f) \in T_M\mathcal{O}^{d\times r}$ # search direction (Equation 34)
  while $f(r_M(\beta\pi_M(-\nabla_M f)))$ is not sufficiently smaller than $f(M)$ **do**
    adjust step size β # line search (using retraction, Equation 35)
  end while
  $M \leftarrow r_M(\beta\pi_M(-\nabla_M f))$ # iterate
end while
return (local) minima M^* of f.

this projection in one step, for example by projecting the free gradient directly onto the manifold. Firstly, while there is a rich literature of such 'one-step' projected gradient methods (Bertsekas, 1976), convergence guarantees only exist for convex constraint sets. Indeed, all matrix manifolds we have discussed are nonconvex (except the trivial $\mathbb{R}^{d\times r}$). The theory of convergence for nonconvex manifolds requires this two-step procedure. Secondly, in our empirical experience, while a one-step projection method does often converge, that convergence is typically much slower than Algorithm 1.

This basic algorithm is extended in two ways: first, the constraint manifold $\mathcal{M}$ is taken to be the Grassmann manifold or some other manifold structure (like the product of Stiefel manifolds, as in CCA above); and second, conjugate gradient methods or second-order optimization techniques can be similarly adapted to the setting of matrix manifolds. Beyond these steps, understanding optimization over matrix manifolds in full generality requires topological and differential geometric machinery that is beyond the scope of this work. All of these topics are discussed in the key reference to this appendix (Absil et al., 2008), as well as the literature cited throughout this paper.

References

T. E. Abrudan, J. Eriksson, and V. Koivunen. Steepest descent algorithms for optimization under unitary matrix constraint. *IEEE Transactions on Signal Processing*, 56:1134–1147, 2008.

P. Absil and J. Malick. Projection-like retractions on matrix manifolds. *SIAM Journal on Optimization*, 22(1):135–158, 2012.

P. A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, Princeton, 2008.

K. P. Adragni and D. Cook. Sufficient dimension reduction and prediction in regression. *Philosophical Transactions of the Royal Society A*, 367:4385–4405, 2009.

S. Amari. Natural gradient learning for over and under complete bases in ICA. *Neural Computation*, 11:1875–1883, 1999.

A. Baccini, P. Besse, and A. de Faguerolles. A L1-norm PCA and heuristic approach. In *Proceedings of the International Conference on Ordinal and Symbolic Data Analysis*, pages 359–368, 1996.

F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Convex optimization with sparsity-inducing norms. *Optimization for Machine Learning*, pages 19–53, 2011.

G. H. Bakır, A. Gretton, M. Franz, and B. Schölkopf. Multivariate regression via Stiefel manifold constraints. In *Pattern Recognition*, pages 262–269. Springer, 2004.

A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall. Learning distance functions using equivalence relations. In *Proceedings of the International Conference on Machine Learning*, volume 3, pages 11–18, 2003.

M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

D. P. Bertsekas. On the goldstein-levitin-polyak gradient projection method. *IEEE Transactions on Automatic Control*, 21(2):174–184, 1976.

C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, New York, 2006.

I. Borg and P. J. Groenen. *Modern Multidimensional Scaling: Theory and Applications*. Springer Verlag, 2005.

M. Borga, T. Landelius, and H. Knutsson. A unified approach to PCA, PLS, MLR, and CCA. *Technical Report*, 1997.

N. Boumal, B. Mishra, P. Absil, and R. Sepulchre. Manopt, a matlab toolbox for optimization on manifolds. *Journal of Machine Learning Research*, 15:1455–1459, 2014.

S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations & Trends in Machine Learning*, 3(1):1–122, 2011.

A. Bray and D. Martinez. Kernel-based extraction of slow features: Complex cells learn disparity and translation invariance from natural images. In *Advances in Neural Information Processing Systems*, pages 253–260, 2002.

W. Brendel, R. Romo, and C. K. Machens. Demixed principal component analysis. In *Advances in Neural Information Processing Systems*, pages 2654–2662, 2011.

W. Buntine. Variational extensions to EM and multinomial PCA. In *Proceedings of the European Conference on Machine Learning*, 2002.

C. J. C. Burges. Dimension reduction: a guided tour. *Foundations & Trends in Machine Learning*, 2(4):275–365, 2010.

E. J. Candes, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of the ACM*, 58(3):11:1–11:37, 2011.

G. Chechik, U. Shalit, V. Sharma, and S. Bengio. An online algorithm for large scale image similarity learning. In *Advances in Neural Information Processing Systems*, pages 306–314, 2009.

V. Choulakian. L1-norm projection pursuit principal component analysis. *Computational Statistics and Data Analysis*, 50(6):1441–1451, 2006.

M. M. Churchland, J. P. Cunningham, M. T. Kaufman, J. D. Foster, P. Nuyujukian, S. I. Ryu, and K. V. Shenoy. Neural population dynamics during reaching. *Nature*, 487:51–56, 2012.

R. R. Coifman and S. Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 21(1):5–30, 2006.

M. Collins, S. Dasgupta, and R. E. Schapire. A generalization of principal component analysis to the exponential family. In *Advances in Neural Information Processing Systems*, 2002.

T. F. Cox and M. A. Cox. *Multidimensional scaling*, volume 88. CRC Press, 2001.

J. P. Cunningham and B. M. Yu. Dimensionality reduction for large-scale neural recordings. *Nature Neuroscience*, 17:1500–1509, 2014.

A d'Aspremont, L. El Ghaoui, M. I. Jordan, and G. R. Lanckriet. A direct formulation for sparse PCA using semidefinite programming. *SIAM Review*, 49:434–448, 2007.

A d'Aspremont, F. R. Bach, and L. El Ghaoui. Optimal solutions for sparse principal component analysis. *Journal of Machine Learning Research*, 9:1269–1294, 2008.

F. De la Torre. A least-squares framework for component analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(6):1041–1055, 2012.

D. De Ridder, R. P. W. Duin, and J. Kittler. Texture description by independent components. In *Structural, Syntactic, and Statistical Pattern Recognition*, pages 587–596. Springer, 2002.

S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm (with discussion). *Journal of the Royal Statistical Society, Series B*, 39:1–38, 1977.

M. Der and L. K. Saul. Latent coincidence analysis: a hidden variable model for distance metric learning. In *Advances in Neural Information Processing Systems*, pages 3230–3238, 2012.

P. Diaconis and D. Freedman. Asymptotics of graphical projection pursuit. *The Annals of Statistics*, pages 793–815, 1984.

C. Eckart and G. Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1:211–218, 1936.

A. Edelman, T. A. Arias, and S. T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM Journal of Matrix Analysis and Applications*, 1998.

K. Fan and A. J. Hoffman. Some metric inequalities in the space of matrices. *Proceedings of the American Mathematical Society*, 6:111–116, 1955.

S. Fiori. Quasi-geodesic neural learning algorithms over the orthogonal group: a tutorial. *Journal of Machine Learning Research*, 6:743–781, 2005.

R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188, 1936.

J. H. Friedman. Exploratory projection pursuit. *Journal of the American Statistical Association*, 82(397):249–266, 1987.

K. Fukumizu, F. R. Bach, and M. I. Jordan. Dimensionality reduction for supervised learning with reproducing kernel hilbert spaces. *Journal of Machine Learning Research*, 5:73–99, 2004.

K. Fukumizu, F. R. Bach, and M. I. Jordan. Kernel dimension reduction in regression. *The Annals of Statistics*, pages 1871–1905, 2009.

K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic press, 1990.

D. Gabay. Minimizing a differentiable function over a differentiable manifold. *The Journal of Optimization Theory and Applications*, 37(2):177–219, 1982.

J. S. Galpin and D. M. Hawkins. Methods of L1 estimation of a covariance matrix. *Computational Statistics and Data Analysis*, 5:305–319, 1987.

A. Globerson and S. T. Roweis. Metric learning by collapsing classes. In *Advances in Neural Information Processing Systems*, pages 451–458, 2005.

J. Goldberger, S. T. Roweis, G. Hinton, and R. Salakhutdinov. Neighbourhood components analysis. In *Advances in Neural Information Processing Systems*, 2004.

G. H. Golub and C. F. Van Loan. *Matrix Computations, 3rd edition*. Hopkins University Press, Baltimore, 1996.

A. Gretton, O. Bousquet, A. Smola, and B. Schölkopf. Measuring statistical dependence with Hilbert-Schmidt norms. In *Algorithmic Learning Theory*, pages 63–77. Springer, 2005.

A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(1):723–773, 2012.

D. R. Hardoon and J. Shawe-Taylor. Convergence analysis of kernel canonical correlation analysis: theory and practice. *Machine Learning*, 74(1):23–38, 2009.

D. R. Hardoon, S. Szedmak, and J Shawe-Taylor. Canonical correlation analysis: an overview with application to learning methods. *Neural Computation*, 16(12):2639–2664, 2004.

T. Hastie, R. Tibshirani, and J. Friedman. *Elements of Statistical Learning, 2nd Edition*. Cambridge University Press, Cambridge, UK, 2008.

X. He and P. Niyogi. Locality preserving projections. In *Advances in Neural Information Processing Systems*, volume 16, page 153, 2004.

X. He, D. Cai, S. Yan, and H. Zhang. Neighborhood preserving embedding. In *IEEE International Conference on Computer Vision*, volume 2, pages 1208–1213, 2005.

N. J. Higham. Matrix nearness problems and applications. In *Applications of Matrix Theory*, pages 1–27. Oxford University Press, 1989.

H. Hotelling. Relations between two sets of variates. *Biometrika*, 28:321–377, 1936.

P. J. Huber. Projection pursuit. *The Annals of Statistics*, pages 435–475, 1985.

A. Hyvarinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. John Wiley and Sons, 2001.

M. Joho, H. Mathis, and R. H. Lambert. Overdetermined blind source separation: Using more sensors than source signals in a noisy mixture. In *Independent Component Analysis and Blind Signal Separation*, pages 81–86, 2000.

M. Journee, Y. Nesterov, P. Richtarik, and R. Sepulchre. Generalized power method for sparse principal component analysis. *Journal of Machine Learning Research*, 11:517–553, 2010.

R. E. Kalman. A new approach to linear filtering and prediction problems. *The Journal of Basic Engineering*, 82:35–45, 1960.

T. Kaneko, S. Fiori, and T. Tanaka. Empirical arithmetic averaging over the compact stiefel manifold. *IEEE Transactions on Signal Processing*, 61(4):883–894, 2013.

Y. H. Kao and B. Van Roy. Learning a factor model via regularized PCA. *Machine Learning*, 91(279-303), 2013.

B. Kulis. Metric learning: A survey. *Foundations & Trends in Machine Learning*, 5(4): 287–364, 2012.

R. Larsen. Decomposition using maximum autocorrelation factors. *Journal of Chemometrics*, 16:427–435, 2002.

N. D. Lawrence. A unifying probabilistic perspective for spectral dimensionality reduction: insights and new models. *Journal of Machine Learning Research*, 13(1):1609–1638, 2012.

D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, 1999.

J. F. Li and X. Y. Hu. Procrustes problems and associated approximation problems for matrices with k-involutory symmetries. *Linear Algebra and its Applications*, 434:820–829, 2011.

K. C. Li. Sliced inverse regression for dimension reduction. *Journal of the American Statistical Association*, 86(414):316–327, 1991.

D. Luenberger. The gradient projection method along geodesics. *Management Science*, 18 (11), 1972.

J. H. Manton. Optimization algorithms exploiting unitary constraints. *IEEE Transactions on Signal Processing*, 50:635–650, 2002.

J. H. Manton. On the various generalizations of optimization algorithms to manifolds. *Proceedings of Mathematical Theory of Network and Systems*, 2004.

K. V. Mardia, J. T. Kent, and J. M. Bibby. *Multivariate Analysis*. Probability and mathematical statistics. Academic Press, 1979.

S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K. R. Müller. Fisher discriminant analysis with kernels. In *Proceedings of the IEEE Signal Processing Society*, pages 41–48, 1999.

L. Mirsky. Symmetric gauge functions and unitarily invariant norms. *Quarterly Journal of Mathematics*, 11:80–89, 1960.

A. Mnih and R. Salakhutdinov. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems*, pages 1257–1264, 2007.

S. Mohamed, K. Heller, and Z. Ghahramani. Bayesian exponential family PCA. In *Advances in Neural Information Processing Systems*, 2008.

R. J. Muirhead. *Aspects of Multivariate Statistical Theory, 2nd Edition*. Wiley, 2005.

J. Nilsson, F. Sha, and M. I. Jordan. Regression on manifolds using kernel dimension reduction. In *Proceedings of the International Conference on Machine Learning*, pages 697–704, 2007.

Y. Nishimori and S. Akaho. Learning algorithms utilizing quasi-geodesic flows on the Stiefel manifold. *Neurocomputing*, 67:106–135, 2005.

K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2:559–572, 1901.

J. Peltonen, J. Goldberger, and S. Kaski. Fast semi-supervised discriminative component analysis. In *Proceeding of the IEEE Workshop on Machine Learning for Signal Processing*, pages 312–317. IEEE, 2007.

C. R. Rao. The utilization of multiple measurements in problems of biological classification. *Journal of the Royal Statistical Society, Series B*, 10(2):159–203, 1948.

C. E. Rasmussen and C.K.I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, 2006.

J. D. Rennie and N. Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *Proceedings of the International Conference on Machine Learning*, pages 713–719, 2005.

S. T. Roweis. EM algorithms for PCA and sensible PCA. In *Advances in Neural Information Processing Systems*, 1997.

S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, December 2000.

E. Rubinshtein and A. Srivastava. Optimal linear projections for enhancing desired data statistics. *Statistical Computing*, 20:267–282, 2010.

A. Ruhe. Closest normal matrix finally found! *Technical Report, University of Goteberg*, 1987.

B. Schölkopf, A. Smola, and R. K. Muller. Kernel principal component analysis. *Advances in Kernel Methods: Support Vector Learning*, pages 327–352, 1999.

P. H. Schonemann. A generalized solution of the orthogonal Procrustes problem. *Psychometrika*, 31:1–10, 1966.

C. Shen, H. Li, and M. J. Brooks. A convex programming approach to the trace quotient problem. In *Proceedings of the Asian Conference on Computer Vision*, pages 227–235. Springer, 2007.

C. Spearman. General intelligence, objectively determined and measured. *American Journal of Psychology*, 15:201–293, 1904.

N. Srebro and T. S. Jaakkola. Weighted low-rank approximations. *Proceedings of the International Conference on Machine Learning*, pages 720–727, 2003.

N. Srebro, J. Rennie, and T. S. Jaakkola. Maximum-margin matrix factorization. In *Advances in Neural Information Processing Systems*, pages 1329–1336, 2004.

A. Srivastava and X. Liu. Tools for application-driven linear dimension reduction. *Neurocomputing*, 67:136–160, 2005.

J. V. Stone and J. Porrill. Undercomplete independent component analysis for signal separation and dimension reduction. *Technical Report*, 1998.

M. Stone and R. J. Brooks. Continuum regression: cross-validated sequentially constructed prediction embracing ordinary least squares, partial least squares and principal components regression. *Journal of the Royal Statistical Society, Series B*, pages 237–269, 1990.

L. Sun, S. Ji, and J. Ye. A least squares formulation for a class of generalized eigenvalue problems in machine learning. In *Proceedings of the International Conference on Machine Learning*, pages 977–984. ACM, 2009.

P. Switzer and A. A. Green. Min/max autocorrelation factors for multivariate spatial imagery. *Technical Report, Stanford University*, 1984.

J. B. Tenenbaum, V. deSilva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, December 2000.

C. M. Theobald. An inequality with application to multivariate analysis. *Biometrika*, 62 (2):461–466, 1975.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1996.

N. H. Timm. *Applied Multivariate Analysis*. Springer, 2002.

M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society, Series B*, 61(3):611–622, 1999.

W. S. Torgerson. Multidimensional scaling: I. theory and method. *Psychometrika*, 17(4): 401–419, 1952.

L. Torresani and K. Lee. Large margin component analysis. In *Advances in Neural Information Processing Systems*, pages 1385–1392, 2006.

R. Turner and M. Sahani. A maximum-likelihood interpretation for slow feature analysis. *Neural Computation*, 19(4):1022–1038, 2007.

M. O. Ulfarsson and V. Solo. Sparse variable PCA using geodesic steepest descent. *IEEE Transactions on Signal Processing*, 56:5823–5832, 2008.

L. J. P. Van der Maaten, E. O. Postma, and H. J. Van den Herik. Dimensionality reduction: A comparative review. *Tilburg University Technical Report, TiCC-TR 2009-005*, 2009.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(2579-2605):85, 2008.

K. R. Varshney and A. S. Willsky. Linear dimensionality reduction for margin-based classification: high-dimensional data and sensor networks. *IEEE Transactions on Signal Processing*, 59:2496–2512, 2011.

M. Wang, F. Sha, and M. I. Jordan. Unsupervised kernel dimension reduction. In *Advances in Neural Information Processing Systems*, pages 2379–2387, 2010.

K. Q. Weinberger and L. K. Saul. Unsupervised learning of image manifolds by semidefinite programming. *International Journal of Computer Vision*, 70(1):77–90, 2006.

K. Q. Weinberger and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10:207–244, 2009.

K. Q. Weinberger, J. Blitzer, and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. In *Advances in Neural Information Processing Systems*, pages 1473–1480, 2005.

M. Welling, F. Agakov, and C. K. I. Williams. Extreme component analysis. In *Advances in Neural Information Processing Systems*, 2003.

M. Welling, R. S. Zemel, and G. E. Hinton. Probabilistic sequential independent components analysis. In *IEEE Transactions on Neural Networks*, 2004.

C. K. I. Williams. On a connection between kernel PCA and metric multidimensional scaling. *Machine Learning*, 46(1-3):11–19, 2002.

C. K. I. Williams and F. Agakov. Products of Gaussians and probabilistic minor component analysis. *Neural Computation*, 14(5):1169–1182, 2002.

L. Wiskott. Slow feature analysis: A theoretical analysis of optimal free responses. *Neural Computation*, 15(9):2147–2177, 2003.

L. Wiskott and T. Sejnowski. Slow feature analysis: unsupervised learning of invariances. *Neural Computation*, 14(4):715–770, 2002.

E. P. Xing, M. I. Jordan, S. Russell, and A. Y. Ng. Distance metric learning with application to clustering with side-information. In *Advances in Neural Information Processing Systems*, pages 505–512, 2002.

S. Yan and X. Tang. Trace quotient problems revisited. In *Proceedings of the European Conference on Computer Vision*, pages 232–244. Springer, 2006.

L. Yang. An overview of distance metric learning. *Proceedings of Computer Vision and Pattern Recognition*, 7, 2007.

L. Yang and R. Jin. Distance metric learning: A comprehensive survey. *Michigan State Universiy Technical Report*, 2006.

S. Yu, K. Yu, V. Tresp, H. P. Kriegel, and M. Wu. Supervised probabilistic principal component analysis. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, pages 464–473, 2006.

L. Zhang, A. Cichocki, and S. Amari. Natural gradient algorithm for blind separation of overdetermined mixture with additive noise. *IEEE Signal Processing Letters*, 6(11): 293–295, 1999.

D. Zhao, Z. Lin, and X. Tang. Laplacian PCA and its applications. In *Proceedings of the International Conference on Computer Vision*, pages 1–8, 2007.

H. Zou, T. Hastie, and R. Tibshirani. Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, 15(2):265–286, 2006.

Journal of Machine Learning Research 16 (2015) 2901-2907 Submitted 9/14; Revised 2/15; Published 12/15

The Randomized Causation Coefficient

David Lopez-Paz* DAVID@LOPEZPAZ.ORG
Max-Planck-Institute for Intelligent Systems,
Spemannstrasse 38, 72076 Tübingen, Germany

Krikamol Muandet KRIKAMOL@TUEBINGEN.MPG.DE
Max-Planck-Institute for Intelligent Systems,
Spemannstrasse 38, 72076 Tübingen, Germany

Benjamin Recht BRECHT@BERKELEY.EDU
Department of EECS, University of California Berkeley,
387 Soda Hall, Berkeley, CA 94720

Editor: Isabelle Guyon and Alexander Statnikov

Abstract

We are interested in learning causal relationships between pairs of random variables, purely from observational data. To effectively address this task, the state-of-the-art relies on strong assumptions on the mechanisms mapping causes to effects, such as invertibility or the existence of additive noise, which only hold in limited situations. On the contrary, this short paper proposes to *learn* how to perform causal inference directly from data, without the need of feature engineering. In particular, we pose causality as a kernel mean embedding classification problem, where inputs are samples from arbitrary probability distributions on pairs of random variables, and labels are types of causal relationships. We validate the performance of our method on synthetic and real-world data against the state-of-the-art. Moreover, we submitted our algorithm to the ChaLearn's "Fast Causation Coefficient Challenge" competition, with which we won the fastest code prize and ranked third in the overall leaderboard.

Keywords: causality, cause-effect inference, kernel mean embeddings, random features

1. Introduction

According to Reichenbach's common cause principle (Reichenbach, 1956), the dependence between two random variables X and Y implies that either X causes Y (denoted by $X \to Y$), or that Y causes X (denoted by $Y \to X$), or that X and Y have a common cause. In this note, we are interested in distinguishing between these three possibilities by using samples drawn from the joint probability distribution P on (X, Y).

Two of the most successful approaches to tackle this problem are the information geometric causal inference method (Daniusis et al., 2012; Janzing et al., 2014), and the additive noise model (Hoyer et al., 2009; Peters et al., 2014). First, the Information Geometric Causal Inference (IGCI) is designed to infer causal relationships between variables related by invertible, noiseless relationships. In particular, assume that there exists a pair of functions or *mapping mechanisms* f and g such that $Y = f(X)$ and $X = g(Y)$. The IGCI method

*. This project was conceived while DLP was visiting BR at University of California, Berkeley.

decides that $X \to Y$ if $\rho(P(X), |\log(f'(X))|) < \rho(P(Y), |\log(g'(Y))|)$, where ρ denotes Pearson's correlation coefficient. IGCI decides $Y \to X$ if the opposite inequality holds, and abstains otherwise. The assumption here is that the cause random variable is independently generated from the mapping mechanism; therefore it is unlikely to find correlations between the density of the former and the slope of the latter. Second, the additive noise model (ANM) assumes that the effect variable is equal to a nonlinear transformation of the cause variable plus some independent random noise, i.e., $Y = f(X) + N_Y$. If $X \perp\!\!\!\perp N_Y$, then there exists no model of the form $X = g(Y) + N_X$ for which $Y \perp\!\!\!\perp N_X$. As a result, one can find the causal direction by performing independence test between the input variable and residual variable in both directions. Specifically, the algorithm will conclude that $X \to Y$ if the pair of random variables (X, N_Y) are independent but the pair (Y, N_X) is not. The algorithm will conclude $Y \to X$ if the opposite claim is true, and abstain otherwise. The additive noise model has been extended to study post-nonlinear models of the form $Y = h(f(X) + N_Y)$, where h a monotone function (Zhang and Hyvärinen, 2009). The consistency of causal inference under the additive noise model was established by Kpotufe et al. (2013) under some technical assumptions.

As it becomes apparent from the previous exposition, there is a lack of a general method to infer causality without assuming strong knowledge about the underlying causal mechanism. Moreover, it is desirable to readily extend inference to other new model hypotheses without incurring in the development of a new, specific algorithm. Motivated by this issue, we raise the question:

Is it possible to automatically learn patterns revealing causal relationships between random variables from large amounts of labeled data?

2. Learning to Learn Causal Inference

Unlike the methods described above, we propose a *data-driven approach* to build a flexible causal inference engine. To do so, we assume access to some set of pairs $\{(S_i, l_i)\}_{i=1}^n$, where the sample $S_i = \{(x_{ij}, y_{ij})\}_{j=1}^{n_i}$ are drawn i.i.d. from the joint distribution P_i of the two random variables X_i and Y_i, which obey the causal relationship denoted by the label l_i. To simplify exposition, the labels $l_i = 1$ denotes $X \to Y$ and $l_i = -1$ stands for $Y \to X$. Using these data, we build a causal inference algorithm in two steps. First, an m-dimensional feature vector $\mathbf{m}_i$ is extracted from each sample S_i, to meaningfully represent the corresponding distribution P_i. Second, we use the set $\{(\mathbf{m}_i, l_i)\}_{i=1}^n$ to train a binary classifier, later used to predict the causal relationship between previously unseen pairs of random variables. This framework can be straightforwardly extended to also infer the "common cause" and "independence" cases, by introducing two extra labels.

Our setup is fundamentally different from the standard classification problem in the sense that the inputs to the learners are samples from probability distributions, rather than real-valued vectors of features (Muandet et al., 2012; Szabó et al., 2014). In particular, we place two assumptions. First, the existence of a *Mother distribution* $\mathcal{M}(\mathcal{P}, \{-1, +1\})$ from which all paired probability distributions $P_i \in \mathcal{P}$ on (X_i, Y_i) and causal labels $l_i \in \{-1, +1\}$ are sampled, where $\mathcal{P}$ denotes the set of all distributions on two real-valued random variables. Second, the causal relationships l_i can be inferred in most cases from observable properties of

the distributions P_i. While these assumptions may not hold in generality, our experimental evidence suggests their wide applicability in real-world data.

The rest of this paper is organized as follows. Section 3 elaborates on how to extract the $m-$dimensional feature vectors $\mathbf{m}_i$ from each causal sample S_i. Section 4 provides empirical evidence to validate our methods. Section 5 closes the exposition by commenting on future research directions.

3. Featurizing Distributions with Kernel Mean Embeddings

Let P be the probability distribution of some random variable Z taking values in $\mathbb{R}^d$. Then, the *kernel mean embedding* of P associated with the positive definite kernel function k is

$$\mu_k(P) := \int_{\mathbb{R}^d} k(z, \cdot) \mathrm{d}P(z) \in \mathcal{H}_k, \tag{1}$$

where $\mathcal{H}_k$ is the reproducing kernel Hilbert space (RKHS) endowed with the kernel k (Berlinet and Thomas-Agnan, 2004; Smola et al., 2007). A sufficient condition which guarantees the existence of μ_k is that the kernel k is bounded, i.e., $\sup_{z \in \mathcal{Z}} k(z, z) < \infty$. One of the most attractive property of μ_k is that it uniquely determines each distribution P when k is a characteristic kernel (Sriperumbudur et al., 2010). In another words, $\|\mu_k(P) - \mu_k(Q)\|_{\mathcal{H}_k} = 0$ iff $P = Q$. Examples of characteristic kernels include the popular squared-exponential

$$k(z, z') = \exp\left(-\gamma \|z - z'\|_2^2\right), \text{ for } \gamma > 0, \tag{2}$$

which will be used throughout this work.

However, in practice, we do not have access to the true distribution P, and consequently to the true embedding μ_k. Instead, we often have access to a sample $S = \{z_i\}_{i=1}^n$ drawn i.i.d. from P. Then, we can construct the empirical measure $P_S = \frac{1}{n} \sum_{i=1}^n \delta_{(z_i)}$, where $\delta_{(z)}$ is the Dirac mass at z, and estimate (1) by

$$\mu_k(P_S) := \frac{1}{n} \sum_{i=1}^n k(z_i, \cdot) \in \mathcal{H}_k. \tag{3}$$

Though it can be improved (Muandet et al., 2014), the estimator (3) is the most common due to its ease of implementation. We can essentially view (1) and (3) as the feature representations of the distribution P and its sample S, respectively.

For some kernels such as (2), the feature maps (1) and (3) do not have a closed form, or are infinite dimensional. This translates into the need of kernel matrices, which require at least $O(n^2)$ computation. In order to alleviate these burdens, we propose to compute a low-dimensional approximation of (3) using random Fourier features (Rahimi and Recht, 2007). In particular, if the kernel k is shift-invariant, we can exploit Bochner's theorem (Rudin, 1962) to construct a randomized approximation of (3), with form

$$\mu_{k,m}(P_S) = \frac{1}{n} \sum_{i=1}^n \left[\cos(w_1' z_i + b_1), \ldots, \cos(w_m' z_i + b_m)\right]' \in \mathbb{R}^m, \tag{4}$$

where the vectors $w_1, \ldots, w_m \in \mathbb{R}^d$ are sampled from the normalized Fourier transform of k, and $b_1, \ldots, b_m \sim \mathcal{U}(0, 2\pi)$. The squared-exponential kernel in (2) is shift-invariant, and

can be approximated in this fashion when setting $w_i \sim \mathcal{N}(0, 2\gamma I)$. These features can be computed in $O(mn)$ time and stored in $O(1)$ memory. Importantly, the low dimensional representation $\mu_{k,m}$ is amenable for the off-the-shelf use with any standard learning algorithm, and not only kernel-based methods.

Using the assumptions introduced in Section 1, the data $\{(\mathbf{m}_i, l_i)\}_{i=1}^n := \{(\mu_{k,m}(P_{S_i}), l_i)\}_{i=1}^n$ and a binary classifier, we can now pose causal inference as a supervised learning problem.

4. Numerical Simulations

We conduct an array of experiments to test the effectiveness of a simple implementation of the presented causal learning framework[1]. Given the use of random embeddings (4) in our classifier, we term our method the *Randomized Causation Coefficient* (RCC). Throughout our simulations, we featurize each sample $S = \{(x_i, y_i)\}_{i=1}^n$ as

$$\nu(S) = (\mu_{k,m}(P_{S_x}), \mu_{k,m}(P_{S_y}), \mu_{k,m}(P_S)), \tag{5}$$

where the three elements forming (5) stand for the low-dimensional representations (4) of the empirical kernel mean embeddings of $\{x_i\}_{i=1}^n$, $\{y_i\}_{i=1}^n$, and $\{(x_i, y_i)\}_{i=1}^n$, respectively. This representation is motivated by the typical conjecture in causal inference about the existence of asymmetries between the marginal and conditional distributions of causally-related pairs of random variables (Schölkopf et al., 2012). Each of these three embeddings has random features sampled to approximate the sum of three Gaussian kernels (2) with hyper-parameters 0.1γ, γ, and 10γ, where γ is set using the median heuristic. In practice, we set $m = 1000$, and observe no significant improvements when using larger amounts of random features. To classify the embeddings (5) in each of the experiments, we use the random forest implementation from Python's `sklearn-0.16-git`. The number of trees forming the forest is chosen from the set $\{100, 250, 500, 1000, 5000\}$, via cross-validation.

4.1 Tübingen Data

The *Tübingen cause-effect pairs* is a collection of heterogeneous, hand-collected, real-world cause-effect samples[2]. Given the small size of this data set, we resort to the synthesis of some Mother distribution to sample our training data from. To this end, assume that sampling a synthetic cause-effect sample $\hat{S}_i := \{(\hat{x}_{ij}, \hat{y}_{ij})\}_{j=1}^n$ equals the following generative process:

1. A *cause* vector $(\hat{x}_{ij})_{j=1}^n$ is sampled from a mixture of Gaussians with c components. The mixture weights are sampled from $\mathcal{U}(0, 1)$, and normalized to sum to one. The mixture means and standard deviations are sampled from $\mathcal{N}(0, \sigma_1)$, and $\mathcal{N}(0, \sigma_2)$, respectively, accepting only positive standard deviations. The cause vector is standardized.

2. A *noise* vector $(\hat{\epsilon}_{ij})_{j=1}^n$ is sampled from a centered Gaussian, with variance sampled from $\mathcal{U}(0, \sigma_3)$.

1. The source code of our experiments is available at `https://github.com/lopezpaz/causation_learning_theory`.
2. The Tübingen cause-effect pairs data set can be downloaded at `https://webdav.tuebingen.mpg.de/cause-effect/`.

3. The *mapping mechanism* $\hat{f}_i$ is a spline fitted using an uniform grid of d_f elements from $\min((\hat{x}_{ij})_{j=1}^n)$ to $\max((\hat{x}_{ij})_{j=1}^n)$ as inputs, and d_f normally distributed outputs.

4. An *effect* vector is built as $(\hat{y}_{ij} := \hat{f}_i(\hat{x}_{ij}) + \hat{\epsilon}_{ij})_{j=1}^n$, and standardized.

5. Return the cause-effect sample $\hat{S}_i := \{(\hat{x}_{ij}, \hat{y}_{ij})\}_{j=1}^n$.

To choose a $\theta = (c, \sigma_1, \sigma_2, \sigma_3, d_f)$ that best resembles the unlabeled test data, we minimize the distance between the embeddings of N synthetic pairs and the Tübingen samples

$$\arg\min_{\theta} \sum_i \min_{1 \le j \le N} \|\nu(S_i) - \nu(\hat{S}_j)\|_2^2,$$

over $c, d_f \in \{1, \ldots, 10\}$, and $\sigma_1, \sigma_2, \sigma_3 \in \{0, 0.5, 1, \ldots, 5\}$, where the $\hat{S}_j$ is sampled using the generative process described above, the S_i are the Tübingen cause-effect pairs, and ν is as in (5). This strategy can be thought of as transductive learning, since we have access to the test inputs (but not their underlying causal relation) at the training time.

We set $n = 1000$, and $N = 10,000$. Using the generative process described above, and the best found parameter vector $\theta = (3, 2, 2, 2, 5)$, we construct the synthetic training data

$$\{\{\nu(\{(\hat{x}_{ij}, \hat{y}_{ij})\}_{j=1}^n), +1)\}_{i=1}^N,$$
$$\{\nu(\{(\hat{y}_{ij}, \hat{x}_{ij})\}_{j=1}^n), -1)\}_{i=1}^N\},$$

where $\{(\hat{x}_{ij}, \hat{y}_{ij})\}_{j=1}^n = \hat{S}_i$, and train our classifier on it. Figure 1 plots the classification accuracy of RCC, IGCI (Daniusis et al., 2012), and ANM (Mooij et al., 2014) versus the fraction of decisions that the algorithms are forced to make out of the 82 scalar Tüebingen cause-effect pairs. To compare these results to other lower-performing methods, refer to Janzing et al. (2012). Overall, RCC surpasses the state-of-the-art in these data, with a classification accuracy of 81.61% when inferring the causal directions on all pairs. The confidence of RCC is computed using the random forest's output class probabilities.

4.2 ChaLearn's "Fast Causation Coefficient" Challenge

We tested RCC at the ChaLearn's *Fast Causation Coefficient* challenge (Guyon, 2014). We trained a Gradient Boosting Classifier (GBC), with hyper-parameters chosen via a 4-fold cross validation, on the featurizations (5) of the training data. In particular, we built two separate classifiers: a first one to distinguish between causal and non-causal pairs (i.e., $X - Y$ vs $\{X \to Y, X \leftarrow Y\}$), and a second one to distinguish between the two possible causal directions on the causal pairs (i.e., $X \to Y$ vs $X \leftarrow Y$). The final causation coefficient for a given sample S_i was computed as

$$\text{score}(S_i) = p_1(S_i) \cdot (2 \cdot p_2(S_i) - 1),$$

where $p_1(x)$ and $p_2(x)$ are the class probabilities output by the first and the second GBCs, respectively. We found it easier to distinguish between causal and non-causal pairs than to infer the correct direction on the causal pairs.

RCC ranked third in the ChaLearn's "Fast Causation Coefficient Challenge" competition, and was awarded the prize to the fastest running code (Guyon, 2014). At the time of the

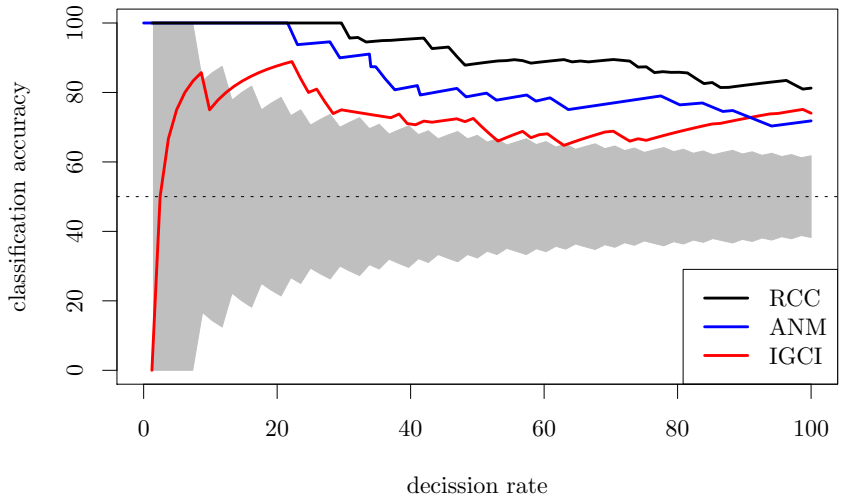


Figure 1: Accuracy of RCC, IGCI and ANM on the Tübingen cause-effect pairs, as a function of decision rate. The grey area depicts accuracies not statistically significant.

competition, we obtained a bidirectional AUC of 0.73 on the test pairs in two minutes of test-time (Guyon, 2014). On the other hand, the winning entry of the competition, which made use of hand-engineered features, took a test-time of 30 minutes, and achieved a bidirectional AUC of 0.82. Interestingly, the performance of IGCI on the $20,000$ training pairs is barely better than random guessing. The computational complexity of the additive noise model (usually implemented as two Gaussian Process regressions followed by two kernel-based independence tests) made it unfeasible to compare it on this data set.

5. Conclusions and Future Research

To conclude, we proposed to *learn how to perform causal inference* between pairs of random variables from observational data, by posing the task as a supervised learning problem. In particular, we introduced an effective and efficient featurization of probability distributions, based on kernel mean embeddings and random Fourier features. Our numerical simulations support the conjecture that patterns revealing causal relationships can be learnt from data.

In light of our encouraging results, we would like to mention four exciting research directions. First, the proposed ideas can be used to learn other *domain-general* statistics, such as measures of dependence (Lopez-Paz et al., 2013). Second, it is important to develop techniques to visualize and interpret the causal features learned by our classifiers. This direction is particularly essential for causal inference as it provides a data-dependent way of discovering new hypothesis on underlying causal mechanism. Third, RCC can be extended to operate not only on pairs, but also sets of random variables, and eventually reconstruct causal DAGs from multivariate data. Finally, one may adapt the distributional learning theory of Szabó et al. (2014) to analyze our randomized, classification setting. For preliminary results on the last two points, we refer the reader to (Lopez-Paz et al., 2015).

References

A. Berlinet and C. Thomas-Agnan. *Reproducing kernel Hilbert spaces in probability and statistics*. Kluwer Academic Publishers, 2004.

P. Daniusis, D. Janzing, J. Mooij, J. Zscheischler, B. Steudel, K. Zhang, and B. Schölkopf. Inferring deterministic causal relations. *UAI*, 2012.

I. Guyon. Chalearn fast causation coefficient challenge, 2014. URL `https://www.codalab.org/competitions/1381`.

P. O. Hoyer, D. Janzing, J. M. Mooij, J. R. Peters, and B. Schölkopf. Nonlinear causal discovery with additive noise models. *NIPS*, 2009.

D. Janzing, J. Mooij, K. Zhang, J. Lemeire, J. Zscheischler, P. Daniušis, B. Steudel, and B. Schölkopf. Information-geometric approach to inferring causal directions. *Artificial Intelligence*, 2012.

D. Janzing, B. Steudel, N. Shajarisales, and B. Schölkopf. Justifying information-geometric causal inference. *arXiv prepring arXiv:1402.2499*, 2014.

S. Kpotufe, E. Sgouritsa, D. Janzing, and B. Schölkopf. Consistency of causal inference under the additive noise model. *ICML*, 2013.

D. Lopez-Paz, P. Hennig, and B. Schölkopf. The Randomized Dependence Coefficient. *NIPS*, 2013.

D. Lopez-Paz, K. Muandet, B. Schölkopf, and I. Tolstikhin. Towards a learning theory of causation. *ICML*, 2015.

J. M. Mooij, J. Peters, D. Janzing, J. Zscheischler, and B. Schölkopf. Distinguishing cause from effect using observational data: methods and benchmarks. *arXiv preprint arXiv:1412.3773*, 2014.

K. Muandet, K. Fukumizu, F. Dinuzzo, and B. Schölkopf. Learning from distributions via support measure machines. *NIPS*, 2012.

K. Muandet, K. Fukumizu, B. Sriperumbudur, A. Gretton, and B. Schölkopf. Kernel mean estimation and Stein effect. *ICML*, 2014.

J. Peters, Joris M. M., D. Janzing, and B. Schölkopf. Causal discovery with continuous additive noise models. *JMLR*, 2014.

A. Rahimi and B. Recht. Random features for large-scale kernel machines. *NIPS*, 2007.

H. Reichenbach. *The direction of time*. Dover, 1956.

W. Rudin. *Fourier analysis on groups*. Wiley, 1962.

B. Schölkopf, D. Janzing, J. Peters, E. Sgouritsa, K. Zhang, and J. M. Mooij. On causal and anticausal learning. In *ICML*, 2012.

A. J. Smola, A. Gretton, L. Song, and B. Schölkopf. A Hilbert space embedding for distributions. In *ALT*. Springer-Verlag, 2007.

B. K. Sriperumbudur, A. Gretton, K. Fukumizu, B. Schölkopf, and G. Lanckriet. Hilbert space embeddings and metrics on probability measures. *JMLR*, 2010.

Z. Szabó, A. Gretton, B. Póczos, and B. Sriperumbudur. Two-stage sampled learning theory on distributions. *arXiv preprint arXiv:1402.1754*, 2014.

K. Zhang and A. Hyvärinen. On the identifiability of the post-nonlinear causal model. *UAI*, 2009.

Journal of Machine Learning Research 16 (2015) 2909-2919 Submitted 9/14; Revised 3/15; Published 12/15

Optimality of Poisson Processes Intensity Learning with Gaussian Processes

Alisa Kirichenko A.KIRICHENKO@UVA.NL
Harry van Zanten HVZANTEN@UVA.NL
Korteweg-de Vries Institute for Mathematics
University of Amsterdam
P.O. Box 94248, 1090 GE Amsterdam, The Netherlands

Editor: Manfred Opper

Abstract

In this paper we provide theoretical support for the so-called "Sigmoidal Gaussian Cox Process" approach to learning the intensity of an inhomogeneous Poisson process on a d-dimensional domain. This method was proposed by Adams, Murray and MacKay (ICML, 2009), who developed a tractable computational approach and showed in simulation and real data experiments that it can work quite satisfactorily. The results presented in the present paper provide theoretical underpinning of the method. In particular, we show how to tune the priors on the hyper parameters of the model in order for the procedure to automatically adapt to the degree of smoothness of the unknown intensity, and to achieve optimal convergence rates.

Keywords: inhomogeneous Poisson process, Bayesian intensity learning, Gaussian process prior, optimal rates, adaptation to smoothness

1. Introduction

Inhomogeneous Poisson processes are widely used models for count and point data in a variety of applied areas. A typical task in applications is to learn the underlying intensity of a Poisson process from a realised point pattern. In this paper we consider nonparametric Bayesian approaches to this problem. These do not assume a specific parametric form of the intensity function and produce posterior distributions which do not only give an estimate of the intensity, for example through the posterior mean or mode, but also give a measure of the remaining uncertainty through the spread of the posterior.

Several papers have explored nonparametric Bayesian approaches in this setting. An early reference is Møller et al. (1998), who study log-Gaussian priors. Gugushvili and Spreij (2013) recently considered Gaussian processes combined with different, non-smooth link functions. Kernel mixtures priors are considered in Kottas and Sansó (2007). Spline-based priors are used in DiMatteo et al. (2001) and Belitser et al. (2013).

The present study is motivated by a method that is not covered by earlier theoretical papers, namely the method of Adams et al. (2009). These authors presented the first approach that is also computationally fully nonparametric in the sense that it does not involve potentially inaccurate finite-dimensional approximations. The method involves a prior on the intensity that is a random multiple of a transformed Gaussian process (GP).

Both the hyper parameters of the GP and the multiplicative constant are endowed with priors as well, resulting in a hierarchical Bayes procedure (details in Section 2.3). Simulation experiments and real data examples in Adams et al. (2009) show that the method can give very satisfactory results.

The aim of this paper is to advance the theoretical understanding of the method of Adams et al. (2009), which they termed "Sigmoidal Gaussian Cox Process" (SGCP). It is by now well known both from theory and practice that nonparametric Bayesian methods need to be tuned very carefully to produce good results. An unfortunate choice of the prior or incorrectly tuned hyper parameters can easily result in procedures that give misleading results or that make sub-optimal use of the information in the training data. See for instance the by now classical reference Diaconis and Freedman (1986), or the more recent paper van der Vaart and van Zanten (2011) and the references therein.

A challenge in this problem (and in nonparametric function learning in general) is to devise a procedure that avoids overfitting and underfitting. The difficulty is that the appropriate degree of "smoothing" depends on the (unknown) regularity of the intensity function that produces the data. Indeed, intuitively it is clear that if the function is very smooth then to learn the intensity at a certain location we can borrow more information from neighboring points than if it is very rough. Ideally we want to have a procedure that automatically uses the appropriate degree of smoothing, that is, that *adapts* to regularity.

To address this issue theoretically it is common to take an asymptotic point of view. Specifically, we assume that we have n independent sets of training data, produced by Poisson processes on the d-dimensional domain $S = [0,1]^d$ (say), with the same intensity function $\lambda_0 : S \to [0,\infty)$. We aim to construct the learning procedure such that we achieve an optimal learning rate, irrespective of the regularity level of the intensity. In the problem at hand it is known that if λ_0 has regularity $\beta > 0$, then the best rate that any procedure can achieve is of the order $n^{-\beta/(d+2\beta)}$. This can be made precise in the minimax framework, for instance. For a fixed estimation or learning procedure, one can determine the largest expected loss that is incurred when the true function generating the data is varied over a ball of functions with fixed regularity β, say. This will depend on n and quantifies the worst-case rate of convergence for that fixed estimator for β-regular truths. The minimax rate is obtained by minimising this over all possible estimators. So it is the best convergence rate that any procedure can achieve, uniformly over a ball of functions with fixed regularity β. See, for example, Tsybakov (2009) for a general introduction to the minimax approach and Kutoyants (1998) or Reynaud-Bouret (2003) for minimax results in the context of the Poisson process model that we consider in this paper.

Note that the smoothness degree is unknown to us, so we can not use it in the construction of the procedure, but still we want that the posterior contracts around λ_0 at the rate $n^{-\beta/(d+2\beta)}$, as $n \to \infty$, if λ_0 is β-smooth. In this paper we prove that with appropriate priors on the hyper parameters, the SGCP approach of Adams et al. (2009) attains this optimal rate (up to a logarithmic factor). It does so for every regularity level $\beta > 0$, so it is fully *rate-adaptive*.

Technically the paper uses the mathematical framework for studying contraction rates for Gaussian and conditionally Gaussian priors as developed in van der Vaart and van Zanten (2008a) and van der Vaart and van Zanten (2009). We also use an extended version of a general result for Bayesian inference for 1-dimensional Poisson processes from Belitser et al.

(2013). On a general level the line of reasoning is similar to that of van der Vaart and van Zanten (2009). However, due to the presence of a link function and a random multiplicative constant in the SGCP model (see Section 2 ahead) the results of the latter paper do not apply in the present setting and additional mathematical arguments are required to prove the desired results.

The paper is organised as follows. In Section 2 we describe the Poisson process observation model and the SGCP prior model, which together determine a full hierarchical Bayesian model. The main result about the performance of the SGCP approach is presented and discussed in Section 3. Mathematical proofs are given in Section 4. In Section 5 we make some concluding remarks.

2. The SGCP Model

In this section we describe the observation model and the SGCP prior model for the intensity.

2.1 Observation Model

We assume we observe n independent copies of an inhomogeneous Poisson process on the d-dimensional unit cube $S = [0,1]^d$ (adaptation to other domains is straightforward). We denote these observed data by $N^1, \ldots, N^n$. Formally every N^i is a counting measure on subsets of S. The object of interest is the underlying *intensity function*. This is a (integrable) function $\lambda : [0,1]^d \to [0,\infty)$ with the property that given λ, every N^j is a random counting measure on $[0,1]^d$ such that $N^j(A)$ and $N^j(B)$ are independent if the sets $A, B \subset [0,1]^d$ are disjoint and the number of points $N^j(B)$ falling in the set B has a Poisson distribution with mean $\int_B \lambda(s)\, ds$. If we want to stress that the probabilities and expectations involving the observations N^j depend on λ, we use the notations P_λ and E_λ, respectively. We note that instead of considering observations from n independent Poisson processes with intensity λ, one could equivalently consider observations from a single Poisson process with intensity $n\lambda$.

2.2 Prior Model

The SGCP model introduced in Adams et al. (2009) postulates a-priori that the intensity function λ is of the form

$$\lambda(s) = \lambda^* \sigma(g(s)), \qquad s \in S, \tag{2.1}$$

where $\lambda^* > 0$ is an upper bound on λ, g is a GP indexed by S and σ is the sigmoid, or logistic function on the real line, defined by $\sigma(x) = (1 + e^{-x})^{-1}$. In the computational section of Adams et al. (2009) g is modeled as a GP with squared exponential covariance kernel and zero mean, with a prior on the length scale parameter. The hyper parameter λ^* is endowed with an independent gamma prior.

In the mathematical results presented in this paper we allow a bit more flexibility in the choice of the covariance kernel of the GP, the link function σ and the priors on the hyper parameters. We assume that g is a zero-mean, homogenous GP with covariance kernel given in spectral form by

$$\mathrm{E} g(s)g(t) = \int e^{-i\langle \xi, \ell(t-s)\rangle} \mu(\xi)\, d\xi, \qquad s,t \in S, \tag{2.2}$$

where $\ell > 0$ is an (inverse) length scale parameter and μ is a spectral density on $\mathbb{R}^d$ such that the map $a \mapsto \mu(a\xi)$ on $(0, \infty)$ is decreasing for every $\xi \in \mathbb{R}^d$ and that satisfies

$$\int e^{\delta||\xi||}\mu(d\xi) < \infty \tag{2.3}$$

for some $\delta > 0$ (the Euclidean inner product and norm are denoted by $\langle \cdot, \cdot \rangle$ and $\| \cdot \|$, respectively). Note that, in particular, the centered Gaussian spectral density satisfies this condition and corresponds to the squared exponential kernel

$$\mathrm{E}g(s)g(t) = e^{-\ell^2\|t-s\|^2}.$$

We endow the length scale parameter ℓ with a prior with density p_ℓ on $[0, \infty)$, for which we assume the bounds, for positive constants C_1, D_1, C_2, D_2, nonnegative constants p, q, and every sufficiently large $x > 0$,

$$C_1 x^p \exp(-D_1 x^d \log^q x) \leqslant p_\ell(x) \leqslant C_2 x^p \exp(-D_2 x^d \log^q x). \tag{2.4}$$

This condition is, for instance, satisfied if ℓ^d has a gamma distribution, which is a common choice in practice. Note however that the technical condition (2.4) is only a condition on the tail of the prior on ℓ. On the upper bound λ^* we put a prior satisfying an exponential tail bound. Specifically, we use a positive, continuous prior density p_{λ^*} on $[0, \infty)$ such that for some $c_0, C_0, \kappa > 0$,

$$\int_{\lambda_0}^{\infty} p_{\lambda^*}(x)\, dx \leqslant C_0 e^{-c_0 \lambda_0^\kappa} \tag{2.5}$$

for all $\lambda_0 > 0$. Note that this condition is fulfilled if we place a gamma prior on λ^*. Finally, we use a strictly increasing, infinitely smooth link function $\sigma : \mathbb{R} \to (0, 1)$ in (2.1) that satisfies

$$|\sqrt{\sigma(x)} - \sqrt{\sigma(y)}| \leqslant c|x - y| \tag{2.6}$$

for all $x, y \in \mathbb{R}$. This condition is in particular fulfilled for the sigmoid function employed by Adams et al. (2009). It holds for other link functions as well, for instance for the cdf of the standard normal distribution.

2.3 Full Hierarchical Model

With the assumptions made in the preceding sections in place, the full hierarchical specification of the prior and observation model can then be summarised as follows:

$$\begin{aligned}
\ell &\sim p_\ell \quad \text{(satisfying (2.4))} \\
\lambda^* &\sim p_{\lambda^*} \quad \text{(satisfying (2.5))} \\
g \,|\, \ell, \lambda^* &\sim \text{GP with kernel given by (2.2)–(2.3)} \\
\lambda \,|\, g, \ell, \lambda^* &\sim \text{defined by (2.1), with smooth } \sigma \text{ satisfying (2.6)} \\
N^1, \ldots, N^n \,|\, \lambda, g, \ell, \lambda^* &\sim \text{independent Poisson processes with intensity } \lambda.
\end{aligned}$$

Note that under the prior, several quantities are, by construction, independent. Specifically, ℓ and λ_* are independent, and g and λ^* are independent.

The main results of the paper concern the posterior distribution of the intensity function λ, that is, the conditional $\lambda \,|\, N^1, \ldots, N^n$. Throughout we will denote the prior on λ by Π and the posterior by $\Pi(\cdot \,|\, N^1, \ldots, N^n)$. In this setting Bayes' formula asserts that

$$\Pi(\lambda \in B \,|\, N^1, \ldots, N^n) = \frac{\int_B p(N^1, \ldots, N^n \,|\, \lambda)\, \Pi(d\lambda)}{\int p(N^1, \ldots, N^n \,|\, \lambda)\, \Pi(d\lambda)}, \tag{2.7}$$

where the likelihood is given by

$$p(N^1, \ldots, N^n \,|\, \lambda) = \prod_{i=1}^{n} e^{\int_S \lambda(x) N^i(dx) - \int_S (\lambda(x)-1)\, dx}$$

(see, for instance, Kutoyants, 1998).

3. Main Result

Consider the prior and observations model described in the preceding section and let $\Pi(\cdot \,|\, N^1, \ldots, N^n)$ be the corresponding posterior distribution of the intensity function λ.

The following theorem describes how quickly the posterior distribution contracts around the true intensity λ_0 that generates the data. The rate of contraction depends on the smoothness level of λ_0. This is quantified by assuming that λ_0 belongs to the Hölder space $C^\beta[0,1]^d$ for $\beta > 0$. By definition a function on $[0,1]^d$ belongs to this space if it has partial derivatives up to the order $\lfloor \beta \rfloor$ and if the $\lfloor \beta \rfloor$th order partial derivatives are all Hölder continuous of the order $\beta - \lfloor \beta \rfloor$. Here $\lfloor \beta \rfloor$ denotes the greatest integer strictly smaller than β. The rate of contraction is measured in the L^2-distance between the square root of intensities. This is the natural statistical metric in this problem, as it can be shown that in this setting the Hellinger distance between the models with intensity functions λ_1 and λ_2 is equivalent to $\min\{\|\sqrt{\lambda_1} - \sqrt{\lambda_2}\|_2, 1\}$ (see Belitser et al., 2013). Here $\|f\|_2$ denotes the L^2-norm of a function on $S = [0,1]^d$, that is, $\|f\|_2^2 = \int_S f^2(s)\, ds$.

Theorem 1 *Suppose that $\lambda_0 \in C^\beta[0,1]^d$ for some $\beta > 0$ and that λ_0 is strictly positive. Then for all sufficiently large $M > 0$,*

$$\mathrm{E}_{\lambda_0} \Pi(\lambda : \|\sqrt{\lambda} - \sqrt{\lambda_0}\|_2 \geqslant M n^{-\beta/(d+2\beta)} \log^\rho n | N^1, \ldots, N^n) \to 0 \tag{3.1}$$

as $n \to \infty$, for some $\rho > 0$.

The theorem asserts that if the intensity λ_0 that generates the data is β-smooth, then, asymptotically, all the posterior mass is concentrated in (Hellinger) balls around λ_0 with a radius that is up to a logarithmic factor of the optimal order $n^{-\beta/(d+2\beta)}$. Since the procedure does not use the knowledge of the smoothness level β, this indeed shows that the method is rate-adaptive, that is, the rate of convergence adapts automatically to the degree of smoothness of the true intensity. Let us mention once again that the conditions of the theorem are in particular fulfilled if in (2.1), λ^* is taken gamma, σ is the sigmoid (logistic) function, and g is a squared exponential GP with length scale ℓ, with ℓ^d a gamma variable.

4. Proof of Theorem 1

To prove the theorem we employ an extended version of a result from Belitser et al. (2013) that gives sufficient conditions for having (3.1) in the case $d = 1$, cf. their Theorem 1. Adaptation to the case of a general $d \in \mathbb{N}$ is straightforward. To state the result we need some (standard) notation and terminology. For a set of positive functions $\mathcal{F}$ we write $\sqrt{\mathcal{F}} = \{\sqrt{f}, f \in \mathcal{F}\}$. For $\varepsilon > 0$ and a norm $\|\cdot\|$ on $\mathcal{F}$, let $N(\varepsilon, \mathcal{F}, ||\cdot||)$ be the minimal number of balls of radius ε with respect to norm $\|\cdot\|$ needed to cover $\mathcal{F}$. The uniform norm $\|f\|_\infty$ of a function f on S is defined, as usual, as $\|f\|_\infty = \sup_{s\in S} |f(s)|$. The space of continuous function on S is denoted by $C(S)$. As usual, $a \wedge b = \min\{a, b\}$ and $a \vee b = \max\{a, b\}$.

Let Π now be a general prior on the intensity function λ and let $\Pi(\cdot \,|\, N^1, \ldots, N^n)$ be the corresponding posterior (2.7).

Theorem 2 *Assume that λ_0 is bounded away from 0. Suppose that for positive sequences $\overline{\delta}_n, \delta_n \to 0$ such that $n(\overline{\delta}_n \wedge \delta_n)^2 \to \infty$ as $n \to \infty$ and constants $c_1, c_2 > 0$, it holds that for all $L > 1$, there exist subsets $\mathcal{F}_n \subset C(S)$ and a constant c_3 such that*

$$1 - \Pi(\mathcal{F}_n) \leqslant e^{-Ln\delta_n^2}, \tag{4.1}$$

$$\Pi(\lambda : ||\lambda - \lambda_0||_\infty \leqslant \delta_n) \geqslant c_1 e^{-nc_2\delta_n^2}, \tag{4.2}$$

$$\log N(\overline{\delta}_n, \sqrt{\mathcal{F}_n}, \|\cdot\|_2) \leqslant c_3 n\overline{\delta}_n^2. \tag{4.3}$$

Then for $\varepsilon_n = \overline{\delta}_n \vee \delta_n$ and all sufficiently large $M > 0$,

$$\mathrm{E}_{\lambda_0}\Pi(\lambda : \|\sqrt{\lambda} - \sqrt{\lambda_0}\|_2 \geqslant M\varepsilon_n | N^1, \ldots N^n) \to 0 \tag{4.4}$$

as $n \to \infty$.

We note that this theorem has a form that is commonly encountered in the literature on contraction rates for nonparametric Bayes procedures. The so-called "prior mass condition" (4.2) requires that the prior puts sufficient mass near the true intensity function λ_0 generating the data. The "remaining mass condition" (4.1) and the "entropy condition" (4.3) together require that "most" of the prior mass should be concentrated on so-called "sieves" $\mathcal{F}_n$ that are not too large in terms of their metric entropy. The sieves grow as $n \to \infty$ and in the limit they capture all the posterior mass.

In the subsequent subsections we will show that the prior defined in Section 2.3 fulfills the conditions of this theorem, for $\delta_n = n^{-\beta/(2\beta+d)}(\log n)^{k_1}$ and $\overline{\delta}_n = L_1 n^{-\beta/(2\beta+d)}(\log n)^{(d+1)/2+2k_1}$, with $L_1 > 0$ and $k_1 = ((1 + d) \vee q)/(2 + d/\beta)$. The proofs build on earlier work, especially from van der Vaart and van Zanten (2009), in which results like (4.1)–(4.3) have been derived for GP's like g. Here we extend and adapt these results to deal with the additional link function σ and the prior on the maximum intensity λ^*.

4.1 Prior Mass Condition

In this section we show that with λ^*, σ and g as specified in Section 2.3 and $\lambda_0 \in C^\beta(S)$, we have

$$\mathrm{P}(\|\lambda^*\sigma(g) - \lambda_0\|_\infty \leqslant \delta_n) \geqslant c_1 e^{-nc_2\delta_n^2} \tag{4.5}$$

for constants $c_1, c_2 > 0$ and δ_n as defined above.

The link function σ is strictly increasing and smooth, hence it has a smooth inverse $\sigma^{-1} : (0,1) \to \mathbb{R}$. Define the function w_0 on S by

$$w_0(s) = \sigma^{-1}\Big(\frac{\lambda_0(s)}{2\|\lambda_0\|_\infty}\Big), \qquad s \in S,$$

so that $\lambda_0 = 2\|\lambda_0\|_\infty \sigma(w_0)$. Since the function λ_0 is positive and continuous on the compact set S, it is bounded away from 0 on S, say $\lambda_0 \geqslant a > 0$. It follows that $\lambda_0(s)/2\|\lambda_0\|_\infty$ varies in the compact interval $[a/2||\lambda_0||_\infty, 1/2]$ as s varies in S, hence w_0 inherits the smoothness of λ_0, that is, $w_0 \in C^\beta(S)$.

Now observe that for $\varepsilon > 0$,

$$\begin{aligned}
&\mathrm{P}(\|\lambda^*\sigma(g) - \lambda_0\|_\infty \leqslant 2\varepsilon) \\
&\qquad = \mathrm{P}(\|(\lambda^* - 2\|\lambda_0\|_\infty)\sigma(g) + 2\|\lambda_0\|_\infty(\sigma(g) - \sigma(w_0))\|_\infty \leqslant 2\varepsilon) \\
&\qquad \geqslant \mathrm{P}(|\lambda^* - 2\|\lambda_0\|_\infty| \leqslant \varepsilon)\mathrm{P}(\|\sigma(g) - \sigma(w_0)\|_\infty \leqslant \varepsilon/2\|\lambda_0\|_\infty).
\end{aligned}$$

Since λ^* has a positive, continuous density the first factor on the right is bounded from below by a constant times ε. Since the function $\sqrt{\sigma}$ is Lipschitz by assumption, the second factor is bounded from below by $\mathrm{P}(\|g - w_0\|_\infty \leqslant c\varepsilon)$ for a constant $c > 0$. By Theorem 3.1 in van der Vaart and van Zanten (2009) we have the lower bound

$$\mathrm{P}(\|g - w_0\|_\infty \leqslant \delta_n) \geqslant e^{-n\delta_n^2},$$

with δ_n as specified above. The proof of (4.5) is now easily completed.

4.2 Construction of Sieves

Let $\mathbb{H}^\ell$ be the RKHS of the GP g with covariance (2.2) and let $\mathbb{H}_1^\ell$ be its unit ball (see van der Vaart and van Zanten, 2008b for background on these notions). Let $\mathbb{B}_1$ be the unit ball in $C[0,1]^d$ relative to the uniform norm. Define

$$\mathcal{F}_n = \bigcup_{\lambda \leqslant \lambda_n} \lambda\sigma(\mathcal{G}_n),$$

where

$$\mathcal{G}_n = \left[M_n\sqrt{\frac{r_n}{\gamma_n}}\mathbb{H}_1^{r_n} + \varepsilon_n\mathbb{B}_1\right] \cup \left[\bigcup_{a \leqslant \gamma_n} (M_n\mathbb{H}_1^a) + \varepsilon_n\mathbb{B}_1\right],$$

and λ_n, M_n, γ_n, r_n and ε_n are sequences to be determined later. In the next two subsections we study the metric entropy of the sieves $\mathcal{F}_n$ and the prior mass of their complements.

4.3 Entropy

Since $\sqrt{\sigma}$ is bounded and Lipschitz we have, for $a, b \in [0, \lambda_n]$, some $c > 0$ and $f, g \in \mathcal{G}_n$,

$$\|\sqrt{a\sigma(f)} - \sqrt{b\sigma(g)}\|_\infty \leqslant |\sqrt{a} - \sqrt{b}| + c\sqrt{\lambda_n}\|f - g\|_\infty.$$

Since $|\sqrt{a}-\sqrt{b}| \leq \sqrt{|a-b|}$ for $a, b > 0$, it follows that for $\varepsilon > 0$,

$$N(2\varepsilon\sqrt{\lambda_n}, \sqrt{\mathcal{F}_n}, \|\cdot\|_2) \leqslant N(\varepsilon\sqrt{\lambda_n}, [0,\lambda_n], \sqrt{|\cdot|}) N(\varepsilon/c, \mathcal{G}_n, \|\cdot\|_\infty),$$

and hence

$$\log N(2\varepsilon\sqrt{\lambda_n}, \sqrt{\mathcal{F}_n}, \|\cdot\|_2) \lesssim \log\left(\frac{1}{\varepsilon}\right) + \log N(\varepsilon/c, \mathcal{G}_n, \|\cdot\|_\infty).$$

By formula (5.4) from van der Vaart and van Zanten (2009),

$$\log N(3\varepsilon_n, \mathcal{G}_n, \|\ \cdot\ \|_\infty) \quad \leqslant \quad K r_n^d \left(\log \frac{d^{1/4} M_n^{3/2} \sqrt{2\tau r_n}}{\varepsilon_n^{3/2}}\right)^{1+d} \quad + \quad 2\log \frac{2M_n\sqrt{||\mu||}}{\varepsilon_n},$$

for $\|\mu\|$ the total mass of the spectral measure μ, τ^2 the second moment of μ, a constant $K > 0$, $\gamma_n = \varepsilon_n/(2\tau\sqrt{d}M_n)$, $r_n > A$ for some constant $A > 0$, and given that the following relations hold:

$$d^{1/4} M_n^{3/2} \sqrt{2\tau r_n} > 2\varepsilon_n^{3/2}, \qquad M_n\sqrt{||\mu||} > \varepsilon_n. \tag{4.6}$$

By substituting $\bar{\eta}_n = \varepsilon_n\sqrt{\lambda_n}$ we get that for some constants K_1 and K_2,

$$\log N(2\bar{\eta}_n, \sqrt{\mathcal{F}_n}, \|\cdot\|_2) \lesssim K_1 r_n^d \left(\log \frac{\lambda_n^{3/4} M_n^{3/2} d^{1/4}\sqrt{2\tau r_n}}{\bar{\eta}_n^{3/2}}\right)^{1+d} + K_2 \log \frac{\lambda_n^{1/2} M_n}{\bar{\eta}_n},$$

when $M_n > 1$. In terms of $\bar{\eta}$ the conditions (4.6) can be rewritten as

$$d^{1/4} M_n^{3/2} \lambda_n^{3/4} \sqrt{2\tau r_n} > 2\bar{\eta}_n^{3/2}, \qquad M_n \lambda_n^{1/2}\sqrt{||\mu||} > \bar{\eta}_n. \tag{4.7}$$

So we conclude that we have the entropy bound

$$\log N(\bar{\eta}_n, \sqrt{\mathcal{F}_n}, \|\cdot\|_2) \lesssim n\bar{\eta}_n^2$$

for sequences λ_n, M_n, r_n and $\bar{\eta}_n$ satisfying (4.7) and

$$K_1 r_n^d \left(\log \frac{\lambda_n^{3/4} M_n^{3/2} d^{1/4}\sqrt{2\tau r_n}}{\bar{\eta}_n^{3/2}}\right)^{1+d} < n\bar{\eta}_n^2, \quad K_2 \log \frac{\lambda_n^{1/2} M_n}{\bar{\eta}_n} < n\bar{\eta}_n^2. \tag{4.8}$$

4.4 Remaining Mass

By conditioning we have

$$\begin{aligned} \mathrm{P}(\lambda^*\sigma(g) \notin \mathcal{F}_n) &= \int_0^\infty \mathrm{P}(\lambda\sigma(g) \notin \mathcal{F}_n) p_{\lambda^*}(\lambda)\, d\lambda \\ &\leqslant \int_0^{\lambda_n} \mathrm{P}(\lambda\sigma(g) \notin \mathcal{F}_n) p_{\lambda^*}(\lambda)\, d\lambda + \int_{\lambda_n}^\infty p_{\lambda^*}(\lambda)\, d\lambda. \end{aligned}$$

By (2.5) the second term is bounded by a constant times $\exp(-c_0\lambda_n^\kappa)$. For the first term, note that for $\lambda \leqslant \lambda_n$ we have

$$\lambda^{-1} \bigcup_{\lambda' \leqslant \lambda_n} \lambda'\sigma(\mathcal{G}_n) \supset \sigma(\mathcal{G}_n),$$

hence $\mathrm{P}(\lambda\sigma(g) \notin \mathcal{F}_n) \leqslant \mathrm{P}(g \notin \mathcal{G}_n)$. From (5.3) in van der Vaart and van Zanten (2009) we obtain the bound

$$\mathrm{P}(g \notin \mathcal{G}_n) \leqslant \frac{K_3 r_n^{p-d+1} e^{-D_2 r_n^d \log^q r_n}}{\log^q r_n} + e^{-M_n^2/8},$$

for some $K_3 > 0$, $\varepsilon_n < \varepsilon_0$ for a small constant $\varepsilon_0 > 0$, and M_n, r_n and ε_n satisfying

$$M_n^2 > 16 K_4 r_n^d (\log(r_n/\varepsilon_n))^{1+d}, \qquad r_n > 1, \tag{4.9}$$

where K_4 is some large constant. It follows that $\mathrm{P}(g \notin \mathcal{G}_n)$ is bounded above by a multiple of $\exp(-Ln\widetilde{\eta}_n^2)$ for a given constant L and $\widetilde{\eta}_n = \lambda_n \varepsilon_n$, provided M_n, r_n, γ_n and ε_n satisfy (4.9) and

$$D_2 r_n^d \log^q r_n \geqslant 2Ln\widetilde{\eta}_n^2, \quad r_n^{p-d+1} \leqslant e^{Ln\widetilde{\eta}_n^2}, \quad M_n^2 \geqslant 8Ln\widetilde{\eta}_n^2. \tag{4.10}$$

Note that in terms of $\tilde{\eta}_n$, (4.9) can be rewritten as

$$M_n^2 > 16 K_4 r_n^d (\log(r_n \lambda_n/\widetilde{\eta}_n))^{1+d}, \qquad r_n > 1. \tag{4.11}$$

We conclude that if (4.11),(4.10) holds and

$$c_0 \lambda_n^\kappa > Ln\widetilde{\eta}_n^2, \tag{4.12}$$

then

$$\mathrm{P}(\lambda^* \sigma(g \notin \mathcal{F}_n)) \lesssim e^{-Ln\widetilde{\eta}_n^2}.$$

4.5 Completion of the Proof

In the view of the preceding it only remains to show that $\widetilde{\eta}_n$, $\bar{\eta}_n$, r_n, $M_n > 1$ and λ_n can be chosen such that relations (4.7), (4.8), (4.10), (4.11) and (4.12) hold.

One can see that it is true for $\widetilde{\eta}_n = \delta_n$ and $\bar{\eta}_n = \overline{\delta}_n$ described in the theorem, with r_n, M_n, λ_n as follows:

$$\begin{aligned} r_n &= L_2\, n^{\frac{1}{2\beta+d}} (\log n)^{\frac{2k_1}{d}}, \\ M_n &= L_3\, n^{\frac{d}{2(2\beta+d)}} (\log n)^{\frac{d+1}{2}+2k_1}, \\ \lambda_n &= L_4\, n^{\frac{d}{\kappa(2\beta+d)}} (\log n)^{\frac{4k_1}{\kappa}} \end{aligned}$$

for some large constants $L_2, L_3, L_4 > 0$.

5. Concluding Remarks

We have shown that the SGCP approach to learning intensity functions proposed by Adams et al. (2009) enjoys very favorable theoretical properties, provided the priors on the hyper parameters are chosen appropriately. The result shows there is some flexibility in the construction of the prior. The squared exponential GP may be replaced by other smooth stationary processes, other link functions may be chosen, and there is also a little room in the choice of the priors on the length scale and the multiplicative parameter. This flexibility is limited, however, and although our result only gives upper bounds on the

contraction rate, results like those of Castillo (2008) and van der Vaart and van Zanten (2011) lead us to believe that one might get sub-optimal performance when deviating too much from the conditions that we have imposed. Strictly speaking the matter is open however and additional research is necessary to make this belief precise and to describe the exact boundaries between good and sub-optimal behaviours.

We expect that a number of generalizations of our results are possible. For instance, it should be possible to obtain generalizations to anisotropic smoothness classes and priors as considered in Bhattacharya et al. (2014), and classes of analytic functions as studied in van der Vaart and van Zanten (2009). These generalizations take considerable additional technical work however and are therefore not worked out in this paper. We believe they would not change the general message of the paper.

Acknowledgments

Research supported by the Netherlands Organisation for Scientific Research, NWO.

References

Adams, R. P., Murray, I. and MacKay, D. J. (2009). Tractable nonparametric Bayesian inference in Poisson processes with Gaussian process intensities. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 9–16. ACM.

Belitser, E., Serra, P. and van Zanten, J. H. (2013). Rate-optimal Bayesian intensity smoothing for inhomogeneous Poisson processes. *To appear in J. Statist. Plann. Inference, arXiv:1304.6017* .

Bhattacharya, A., Pati, D. and Dunson, D. (2014). Anisotropic function estimation using multi-bandwidth Gaussian processes. *Ann. Statist.* **42**(1), 352–381.

Castillo, I. (2008). Lower bounds for posterior rates with Gaussian process priors. *Electron. J. Stat.* **2**, 1281–1299.

Diaconis, P. and Freedman, D. (1986). On the consistency of Bayes estimates. *Ann. Statist.* **14**(1), 1–67.

DiMatteo, I., Genovese, C. R. and Kass, R. E. (2001). Bayesian curve-fitting with free-knot splines. *Biometrika* **88**(4), 1055–1071.

Gugushvili, S. and Spreij, P. (2013). A note on non-parametric Bayesian estimation for Poisson point processes. *ArXiv E-prints*.

Kottas, A. and Sansó, B. (2007). Bayesian mixture modeling for spatial Poisson process intensities, with applications to extreme value analysis. *Journal of Statistical Planning and Inference* **137**(10), 3151–3163.

Kutoyants, Y. A. (1998). *Statistical inference for spatial Poisson processes*. Springer.

Møller, J., Syversveen, A. R. and Waagepetersen, R. P. (1998). Log Gaussian Cox processes. *Scandinavian Journal of Statistics* **25**(3), 451–482.

Reynaud-Bouret, P. (2003). Adaptive estimation of the intensity of inhomogeneous Poisson processes via concentration inequalities. *Probab. Theory Related Fields* **126**(1), 103–153.

Tsybakov, A. (2009). *Introduction to Nonparametric Estimation.* Springer Series in Statistics. Springer, New York.

van der Vaart, A. W. and van Zanten, J. H. (2008a). Rates of contraction of posterior distributions based on Gaussian process priors. *Ann. Statist.* **36**(3), 1435–1463.

van der Vaart, A. W. and van Zanten, J. H. (2008b). Reproducing kernel Hilbert spaces of Gaussian priors. *IMS Collections* **3**, 200–222.

van der Vaart, A. W. and van Zanten, J. H. (2009). Adaptive Bayesian estimation using a Gaussian random field with inverse gamma bandwidth. *Ann. Statist.* **37**(5B), 2655–2675.

van der Vaart, A. W. and van Zanten, J. H. (2011). Information rates of nonparametric Gaussian process methods. *J. Mach. Learn. Res.* **12**, 2095–2119.

Journal of Machine Learning Research 16 (2015) 2921-2947 Submitted 3/14; Revised 8/14; Published 12/15

Combination of Feature Engineering and Ranking Models for Paper-Author Identification in KDD Cup 2013

Chun-Liang Li R01922001@NTU.EDU.TW
Yu-Chuan Su R01922159@NTU.EDU.TW
Ting-Wei Lin R01944011@NTU.EDU.TW
Cheng-Hao Tsai R01922025@NTU.EDU.TW
Wei-Cheng Chang B99902019@NTU.EDU.TW
Kuan-Hao Huang B99902059@NTU.EDU.TW
Tzu-Ming Kuo B99902073@NTU.EDU.TW
Shan-Wei Lin B99902023@NTU.EDU.TW
Young-San Lin B97902055@NTU.EDU.TW
Yu-Chen Lu B98902105@NTU.EDU.TW
Chun-Pai Yang B99902109@NTU.EDU.TW
Cheng-Xia Chang R01944041@NTU.EDU.TW
Wei-Sheng Chin D01944006@NTU.EDU.TW
Yu-Chin Juan R01922136@NTU.EDU.TW
Hsiao-Yu Tung B98901044@NTU.EDU.TW
Jui-Pin Wang R01922165@NTU.EDU.TW
Cheng-Kuang Wei B98901037@NTU.EDU.TW
Felix Wu B99902090@NTU.EDU.TW
Tu-Chun Yin D00922023@NTU.EDU.TW
Tong Yu R01922141@NTU.EDU.TW
Yong Zhuang R01922139@NTU.EDU.TW
Shou-de Lin SDLIN@CSIE.NTU.EDU.TW
Hsuan-Tien Lin HTLIN@CSIE.NTU.EDU.TW
Chih-Jen Lin CJLIN@CSIE.NTU.EDU.TW
National Taiwan University
Taipei 106, Taiwan

Editor: Senjuti Basu Roy, Vani Mandava, and Martine De Cock

Abstract

This paper describes the winning solution of team National Taiwan University for track 1 of KDD Cup 2013. The track 1 in KDD Cup 2013 considers the paper-author identification problem, which is to identify whether a paper is truly written by an author. First, we conduct feature engineering to transform the various types of provided text information into 97 features. Second, we train classification and ranking models using these features. Last, we combine our individual models to boost the performance by using results on the internal validation set and the official Valid set. Some effective post-processing techniques have also been proposed. Our solution achieves 0.98259 MAP score and ranks the first place on the private leaderboard of the Test set.

Keywords: paper-author identification, feature generation

1. Introduction

In recent years, different open platforms such as Microsoft Academic Search,[1] Google Scholar,[2] and DBLP[3] have been constructed for providing various papers and authors information for the research community. One of the main challenges of providing this service is, by collecting the information from different sources on the Internet, author profiles may be incorrectly assigned to papers that are not written by them. This situation could be caused by author-name ambiguity, the same name shared by different authors, and the wrong paper-author information from the source. The research problem to address this challenge is called *paper-author identification*, which is to identify which papers are truly written by an author.

We briefly review existing approaches for this problem. Some have modeled it as a link prediction problem in social networks. For example, Sun et al. (2011) introduce Heterogeneous Bibliographic Network, which contains multiple types of nodes, including authors, papers, and topics. The links among these nodes represent different relations between authors and papers. Then several topological features could be extracted from the network to assist supervised learning techniques for link prediction. Sun et al. (2011) systematically extract some heterogeneous-network features and demonstrate that they are more effective than traditional homogeneous-network features.

Sun et al. (2012) generalize the concept of heterogeneous bibliographic networks to general heterogeneous networks. Their model leverages the interaction between different types of nodes to mine more semantic information of the network. Yang et al. (2012) apply probabilistic approaches and explore the temporal information on the network. Their experimental results on co-authorship prediction demonstrate the effectiveness. Lee and Adorna (2012) modify a heterogeneous bibliographic network by highlighting important relations in the network. Kuo et al. (2013) further study the heterogeneous network under the unsupervised settings with aggregate statistics. Besides the link prediction problem, the heterogeneous network has also been applied to other related problems, such as citation prediction (Sun et al., 2011; Yu et al., 2012).

Another problem related to paper-author identification is authorship contribution (Juola, 2006; Stamatatos, 2009). The goal of authorship contribution is to infer the characteristics of authors from given texts. Then we can distinguish the texts written by different authors.

KDD Cup is currently one of the most important data mining competitions. In 2013, track 1 of KDD Cup considers a problem of paper-author identification. The data set is provided by Microsoft Academic Search. Participants are given thousands of authors and their publications. However, for any author, some papers may be wrongly assigned to him/her. Therefore, the goal of this competition is to identify which paper is truly written by an author from the given publications.

The paper describes the winning solution of team National Taiwan University. Our approach treats the problem as a binary classification or ranking problem. Therefore, we conduct feature engineering transforming the given text information into features and then apply the state-of-art binary classification and ranking algorithms. Last, we ensemble

1. http://academic.research.microsoft.com/
2. http://scholar.google.com.tw/
3. http://dblp.uni-trier.de/

several classification models and conduct a post-processing procedure to further boost the performance. According to the announced results, our approach achieves the best result with 0.98259 MAP score.

The paper is organized as follows. Section 2 introduces the track 1 problem of KDD Cup 2013. Section 3 outlines the framework of our approaches. Section 4 describes the approaches to transform the given text information into meaningful features. Section 5 discusses the models that we used. Section 6 describes how we combine different models and post-process the combined result to boost the performance. Finally, we conclude and discuss potential issues in Section 7.

Our implementation is available at `https://github.com/kdd-cup-2013-ntu/track1`. A preliminary version of the paper appeared in the KDD Cup 2013 Workshop (Li et al., 2013).

2. Track 1 of KDD Cup 2013

The data set of track 1 of KDD Cup 2013 (Roy et al., 2013) is provided by Microsoft Academic Search. To address the paper-author identification problem, Microsoft Academic Search provides an interface allowing authors to confirm or delete the papers in their profiles. Confirmation means authors acknowledge they are the authors of the given paper; in contrast, deletion means authors claim that they are not the authors of the given papers (Roy et al., 2013). The data set contains the information about authors and their confirmed/deleted papers. Based on author IDs, the organizers split the data set to three parts, including *Train*, *Valid*, and *Test* sets.

The *Train* set (`Train.csv`) contains 3,739 authors. For each author, the AuthorId, ConfirmedPaperIds, and DeletedPaperIds are provided. The *Valid* set (`Valid.csv`) of 1,486 authors, each with an associated sequence of assigned paper IDs without confirmation or deletion, is for public leaderboard evaluation. The answers (confirmation/deletion) in the *Valid* set were released two weeks before the end of the competition. Participants were allowed to refine their algorithms based on the released answers of the *Valid* set, and were required to submit their models one week before the end of the competition. After the submission, the *Test* set (`Test.csv`) of 2,245 authors was used for private leaderboard evaluation.

In addition to the *Train* set, the following information is also provided.

- `Author.csv` contains author names and their affiliations.
- `Paper.csv` contains paper titles, years, conference IDs, journal IDs, and keywords.
- `PaperAuthor.csv` contains paper IDs, author IDs, author names, and affiliations.
- `Journal.csv` contains short names, full names, and home page information of journals.
- `Conference.csv` contains short names, full names, and home page information of conferences.

	# of authors	# of papers	# of confirmed papers	# of deleted papers
`Train.csv`	3,739	224,459	224,459	108,794
`Valid.csv`	1,486	86,755	41,024	47,081
`Test.csv`	2,245	129,427	–	–
`Paper.csv`	–	2,257,249	–	–
`Author.csv`	247,203	–	–	–
`PaperAuthor.csv`	2,143,148	2,258,482	–	–

Table 1: Statistics of the given files.

		Mean	Std.	Median	Min	Max	Q_1	Q_3
`Train.csv`	Confirmed	33.02	52.72	15	1	860	5	38
`Train.csv`	Deleted	30.08	107.34	6	1	2933	2	21
`Train.csv`	All	63.09	124.81	28	2	2977	11	68
`Valid.csv`	Confirmed	32.32	56.66	14	1	1324	5	38
`Valid.csv`	Deleted	27.79	78.12	5	1	1872	2	22
`Valid.csv`	All	60.22	103.15	28	2	2048	11	69
`Test.csv`	All	60.45	100.68	28	2	1371	11	68

Table 2: Statistics on the number of papers per author in the data set, where Q_1 and Q_3 are the first and third quartiles, respectively.

Unfortunately, the provided additional data are noisy and have missing values. For instance, `PaperAuthor.csv` contains the relations of authors and papers, but papers may be incorrectly assigned to an author. More statistics of the data are provided in Tables 1 and 2.

The goal of the competition is to predict which given papers are written by the given author. To be more specific, given confirmation and deletion records of authors as the training data (`Train.csv`), participants of the competition must predict which papers in the given paper list of each author in the test data (`Test.csv`) are truly written by him or her. The evaluation criterion is mean average precision (MAP), which is commonly used for ranking problems. Before answers of the *Valid* set were released, each team was allowed to submit their results on the *Valid* set five times per day and MAP scores were shown on the public leaderboard. During the last week of the competition, each team was allowed to submit multiple results on the *Test* set, and select one result for the final standing.

At National Taiwan University, we organized a course for KDD Cup 2013. Our members include three instructors, three TAs, and 18 students. The students were split into six sub-teams. Every week, each sub-team presented their progress and discussed with other sub-teams. The TAs helped to build an internal competition environment such that each sub-team could try their ideas before submitting their results to the competition website. Following the competition rules, the whole team share a single account for submitting results. According to the announced results, our approach achieves the best result on the *Test* set with 0.98259 MAP score.

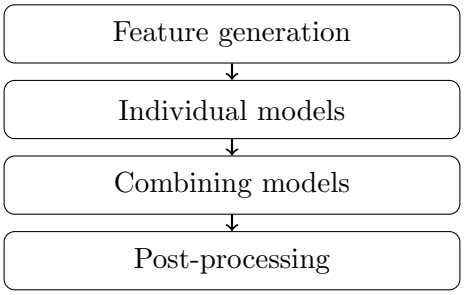


Figure 1: The framework of our approach.

3. Framework

This section first provides the framework of our system. Then we discuss the self-split internal validation set from the *Train* set. The internal validation is not only useful for off-line validating the model performance and combining different models, but also important for avoiding over-fitting the *Valid* set.

3.1 System Overview

We mentioned in Section 1 that we take a supervised learning approach. Our system can be divided into four stages: generating features, training individual models, combining different models, and post-processing as shown in Figure 1. The framework is similar to the one proposed in Yu et al. (2010), which is effective in data-mining applications.

In the first stage, we transform `Train.csv` into a binary classification training set. For each author in `Train.csv`, the list of confirmed papers and deleted papers are provided as described in Section 2. For each paper on the list, we can generate a corresponding author-paper pair, and each pair is treated as a training instance. The confirmation of an author-paper pair is a training instance with label 1; the deletion of of an author-paper pair is a training instance with label -1. We explore different approaches to generate features, which capture various aspects of the given text information.

In the second stage, we mainly employ three models, including Random Forests, Gradient Boosting Decision Tree and LambdaMART. For each individual model, to avoid over-fitting, we carefully conduct the parameter selection by using the internal validation set. In the third stage, we combine the three different models by using results on the internal validation set and the official *Valid* set. In the last stage, we post-process the combined result to further improve the performance by exploiting duplicated information which is not fully utilized by the models.

3.2 Validation Set

A validation set independent from the training set is useful for evaluating models. Given that the answers of the official *Valid* set are not available in the early stage of the competition, we construct an internal validation set for verifying our models. It is also useful to avoid over-fitting leaderboard results on the *Valid* set. In this competition, official *Train*, *Valid*

and *Test* sets are generated by first randomly shuffling authors, and then separate them into three parts with ratio 5:2:3 respectively. Therefore, we randomly split the *Train* set to have 2,670 authors as the internal training set and 1,069 authors as the internal validation set. In our experiments, the MAP score on the internal validation set is usually consistent with the one computed by five-fold cross validation on the official *Train* set.

4. Feature Engineering

To determine the confirmation or deletion of each author-paper pair, we treat each author-paper pair in `Train.csv` as a training instance with label 1 or -1 that represents confirmation or deletion, respectively. We then generate 97 features for each instance and apply the learning algorithms described in Section 5. Subsequently, in describing the feature generation for each author-paper pair, we refer to the author and the paper as the target author and the target paper, respectively.

In this section, we describe our approaches of transforming the given information into features. For the full feature list, please refer to the Appendix.

4.1 Preprocessing

Since many features are based on string matching, we conduct simple preprocessing to clean the data. We first replace the Latin alphabet with the English alphabet, such as replacing ó with o; we also delete some Greek alphabet letters, such as π. Then, we remove stop words in affiliations, titles and keywords, where the stop-word list is obtained from the NLTK package (Bird et al., 2009). Finally, we convert all characters into lowercase before comparison.

4.2 Features Using Author Information

This type of features stems from user profiles, such as user names or affiliations. Based on the information we try to capture, these features can be classified into the following three groups.

4.2.1 Confirmation of Author Profiles

An intuitive method to confirm that a paper is written by a given author is to check whether the name appears in the author section of the paper. However, a more careful setting is to check also the consistency of other information such as affiliations. In the competition, author affiliations are provided in `Author.csv` and `PaperAuthor.csv`. One basic assumption about `Author.csv` and `PaperAuthor.csv` is that `Author.csv` contains the author profiles maintained by Microsoft Academic Search, while the author information in `PaperAuthor.csv` is extracted from the paper without confirmation. The assumption is based on our observation on the given files as well as the online system. When there exists a conflict between `Author.csv` and `PaperAuthor.csv`, the author information in the online system is usually the same as that in `Author.csv`. Therefore, we generate features by comparing author names and affiliations between `Author.csv` and `PaperAuthor.csv`. The comparisons are done by string matching, and various string distances are used as features, including Jaro distance (Jaro, 1989, 1995), Levenshtein distance (Levenshtein,

1966), Jaccard distance (Jaccard, 1901a,b) (of words) and character match ratio. These features are simple but useful; for example, by using only the affiliation Levenshtein distance as a feature, we can achieve 0.94 MAP score on the *Valid* set.

An issue in author-name matching is to handle abbreviated names, which are very common in `PaperAuthor.csv`. In contrast, author names in `Author.csv` are usually in a complete format. The string distance between an abbreviated name and a full name may be large even if the two names are the same. Two different approaches are used to overcome the problem. The first one is to convert all names into an abbreviated format before the comparison; in our approach, the conversion is done by retaining only the last name and first character of first and middle names. The second approach is to split the author name into first, last and middle names, and compare each of them separately. The two approaches are applied independently to obtain different features.

Another challenge of name matching comes from the inconsistency of the name order. There are two main name orders in the provided data, the Western order and the Eastern order. The Western order means that given names precede surnames; in contrast, the Eastern order means that surnames precede given names. While most of the names are in the Western order, names in the Eastern order also frequently appear to cause failed comparisons. Although it is possible to check the name order and transform the Eastern-order names to Western-order ones before comparisons, such checking might be difficult and is prone to error. Instead, two different features are generated for the same distance measure. One assumes that names from `Author.csv` and `PaperAuthor.csv` are in the same name order. The other assumes that names are in the opposite order, so the name order in `Author.csv` is changed before string comparisons. Specifically, the order change is done by exchanging the first word and the last word in the name. However, this setting may wrongly consider two different author names as the same; for example, **Xue Yan** (PID:1224852) and **Yan Xue** (PID:482431) are considered as the same person in the generation of the second feature. Fortunately, because the number of Eastern-order name is relatively small in the data set, our approach still improves the overall performance.

4.2.2 Coauthor Name Matching

Features matching coauthor names are inspired by observing the data set: in many deleted papers, there exist coauthors with names similar to the target author. For example, two authors (174432 and 1363357) of the deleted paper 5633 are the same as the target author **Li Zhang**. Therefore, having such coauthors is an important trait of deleted papers. To capture the information, we take the minimum string distance of names between the target author and his/her coauthors as a feature. Similar to the feature generation in Section 4.2.1, we also need to address the issue of abbreviated names and name orders.

Another problem for matching coauthor names is to decide names for comparison. For a given author identifier, corresponding names may appear in both `Author.csv` and `PaperAuthor.csv`. In fact, multiple names under the same identifier may appear in `PaperAuthor.csv`. These names may be different because of abbreviations, typos or even parsing errors of the Microsoft system. For example, author 1149778 is **Dariusz Adam Ceglarek** in `Author.csv`, while it corresponds to **Dariusz Ceglarek** and **D. Ceglarek** under paper 770630 in `PaperAuthor.csv`. Besides, some authors in `PaperAuthor.csv` do

not appear in `Author.csv`. To handle the problem, multiple features are generated, where each feature is computed by using different combinations of name sources. For instance, the target author name could be from `Author.csv` and `PaperAuthor.csv`, and coauthor names could be from `PaperAuthor.csv`. Then the distances of all possible combinations of the author and each coauthor names from different sources are computed. We select the minimum distance among all possible combinations to represent the name distance between the author and his/her coauthors. We give some examples to illustrate this type of features. One of the features is the maximal Jaro distance between the target author and all coauthors in the target paper. The list of coauthors is from the information in `PaperAuthor.csv`. To extract useful information from the names, we consider different forms of names for computing the distance: full name, abbreviated name, first name, last name and name under the order change (see Section 4.2.1). We also employ other distance measures to obtain more features; see a complete list in Appendix A.1

4.2.3 Author Consistency

Understandably, information in the data set should be consistent across papers and authors. Author-consistency features try to measure such information in author profiles. In particular, we measure the coauthor-affiliation consistency and research-topic consistency as features. Affiliation consistency is based on the assumption that authors with the same affiliation are more likely to co-work on a paper; therefore, we compute the affiliation string distance as well as the number of coauthors with the same affiliation as the target author. Similar to coauthor name matching, the affiliation may come from different sources, so we compute multiple features.

Research-topic consistency assumes that the author should work on related topics across different papers. Although the research topic or field information is not given in the data set, we infer it from the paper titles and keywords. Therefore, we compute the title and keyword similarity between the target paper and other papers of the target author as features.

4.2.4 Missing Value Handling

Missing values cause difficulties in conducting string matching. A common situation in comparing author affiliations or author names is that both strings are empty. The resulting zero string distance wrongly indicates an identical match. As a result, papers with missing values tend to be ranked higher in prediction. To overcome this problem, we consider values other than zero in calculating the distance. If both strings for comparison are empty, we define their Jaro distance as 0.5, Jaccard distance as 0.5 and Levenshtein distance as the average length of the field. Besides, we use some indicators as features; examples include the number of coauthors without affiliation information.

4.3 Features Using Publication Time

Publication-time features are related to the publication year provided in `Paper.csv`. The intuition of these features is that an author can be active in a specific period, and papers written outside this period are likely authored by others. We include several features to capture the publication-time information, such as the exact publication year, publication-time span and publication year differences with other papers of the target author.

Determining whether the provided year is valid is an issue to resolve before we can generate year features. In the data set, some papers' publication years such as 0, -1, and 800190 are obviously invalid. Besides, experiments on the internal validation set show that excluding publication years earlier than 1800 A.D. improves the overall performance. Therefore, we set the valid interval to be between 1800 A.D. and 2013 A.D. and ignore publication years outside the interval.

Removing invalid publication years incurs the missing value problem. To fill the missing year values, we utilize the publication-year information of coauthors. The basic concept is to replace a missing value with the average of the mean publication years of all coauthors of the paper. This average, however, is not computable because coauthors may also have missing information on publication years. An iterative process is used to solve the problem as follows. First, papers with invalid years are ignored and mean of available publication years is calculated for each author. The mean value is then used to fill the missing value of the author. These new values can be incorporated to calculate the new mean value of the publication years. Therefore, the mean publication years and missing values are computed alternatively until convergence. We list the procedure as follows. Please refer to our implementation for detailed steps.

1. Let $\mathcal{P}$ be the set of papers with valid years, and $m_{\mathcal{P}}$ be the map that maps each paper $p \in \mathcal{P}$ to its publication year.

2. Let $\mathcal{A}$ be the set of authors of $\mathcal{P}$ and $m_{\mathcal{A}}$ be the map that maps each author $a \in \mathcal{A}$ to his/her mean publication year calculated based on $m_{\mathcal{P}}$.

3. Let $\mathcal{P}'$ be the set of papers with invalid years, and $m_{\mathcal{P}'}$ be the map that maps each paper $p \in \mathcal{P}'$ to the average of mean publication years of its authors in m_a. If the paper $p \in \mathcal{P}'$ does not have any author in $\mathcal{A}$, the publication year is assigned to 0 in $m_{\mathcal{P}'}$.

4. Let $m_{\mathcal{P}''} = m_{\mathcal{P}'}$.

5. Let $\mathcal{A}'$ be the set of authors having papers in $\mathcal{P} \cup \mathcal{P}'$, and $m_{\mathcal{A}'}$ be the map that maps each author $a \in \mathcal{A}'$ to his/her mean publication year calculated from $m_{\mathcal{P}}$ and $m_{\mathcal{P}'}$.

6. Update $m_{\mathcal{P}'}$ by mapping $p \in \mathcal{P}'$ to the average of mean publication year of its authors according to $m_{\mathcal{A}'}$.

7. If the mean squared differences between years of $m_{p'}$ and $m_{p''}$ is smaller than a given threshold, any zero entry of $m_{p'}$ is replaced by the mean year of m_p and $m_{p'}$. Then stop the procedure and return $\mathcal{P}'$ and $m_{\mathcal{P}'}$.

8. Let $m_{\mathcal{P}''} = m_{\mathcal{P}'}$ and go to step 5.

4.4 Features Using Heterogeneous Bibliographic Networks

Sun et al. (2011) introduce the concept of Heterogeneous Bibliographic Network to capture the different relations between authors and papers, and demonstrate the effectiveness of

link prediction. In this competition, finding whether a paper is written by a given author becomes predicting a link between an author and a paper. According to our study, the relationship between authors and their publications, or coauthors is very useful for linking prediction. This observation is consistent with the claim in Sun et al. (2011). Because such information can be captured by Heterogeneous Bibliographic Network, and by computing certain structures of the network as features, we can obtain the relation from the network to improve the prediction accuracy.

Heterogeneous Bibliographic Network is a graph $G = (V, E)$, where V is the vertex set and E is the edge set. According to the given data, the vertex set $V = \mathcal{P} \cup \mathcal{A} \cup \mathcal{C} \cup \mathcal{J}$ contains the set of papers $\mathcal{P}$, the set of authors $\mathcal{A}$, the set of conferences $\mathcal{C}$ and the set of journals $\mathcal{J}$. The set E consists of two kinds of edges. Based on `PaperAuthor.csv`, if author a_i writes paper p_j, then we create the edge e_{ij}; based on `Papers.csv`, if paper p_m belongs to conference c_n or journal j_n, then we create the edge e_{mn}. Note that, because information in `PaperAuthor.csv` may be incorrect, some links are wrongly generated in the network.

After generating the network, we could extract basic features, such as the number of publications of an author, and the number of total coauthors of an author.

To utilize the network structure, we further define the "path" to describe node relationship. Given the paper-author pair (p_i, a_j), a length-k *meta path* is defined as $(p_i \leftrightarrow v_1 \leftrightarrow \cdots \leftrightarrow v_{k-1} \leftrightarrow a_j)$, where $v_1, \cdots, v_{k-1} \in V$ and $\leftrightarrow$ means two nodes are connected by an edge. Various paths of the graph are extracted as features. In Appendix A.3, we list all kinds of meta paths used to generate features. Although these paths are extracted from the graph structure, they have clear physical meaning and can be interpreted easily. For example, the sixth feature on the list corresponds to the size of the following meta-path set: $S_{mn} = \{(p_m \leftrightarrow j \leftrightarrow \bar{p} \leftrightarrow a_n)\}$, where (p_m, a_n) are given and length-3 meta paths capture all papers of author a_n published in the same journal j as p_m. Take the eighteenth feature as another example. Under given (p_m, a_n), this feature indicates the number of different p_j's on meta paths $(a_n \leftrightarrow p_m \leftrightarrow a_i \leftrightarrow p_j)$. It captures the total number of papers written by coauthors in the target paper.

Further, given an author pair (a_i, a_j), a length-k *pseudo path* is defined as $(a_i \sim a_1 \sim \cdots \sim a_{k-1} \sim a_j)$, where $a_1, \cdots, a_{k-1} \in \mathcal{A}$. Because there is no edge between two author nodes in our network, $\sim$ is a pseudo edge. If author node a_j is reachable from a_i on the network by traversing non-author nodes, then we consider there is a pseudo edge between a_i and a_j. In other words, the pseudo edge describes the possible co-authorship between two authors. By considering the pseudo paths, we can grasp different co-authorship information. The second feature in Appendix A.3 uses the pseudo-edge information directly by computing the number of neighboring a_i's of the target author a_n. This means the number of coauthors of the target author. The pseudo edge is also used implicitly by many other features. For example, the sixteenth feature in Appendix A.3 relies on pseudo edges to identify the coauthors of the target author and then computes the average number of papers of the coauthors.

5. Models

After generating features, we apply classification and ranking methods to train the data set. To enhance the diversity, we explore tree-based classifiers and linear classifiers. The tree-based algorithms including Random Forests (Breiman, 2001), Gradient Boosting Decision Tree (Friedman, 2002), and LambdaMart (Wu et al., 2010). The linear classifier we have studied is RankSVM (Herbrich et al., 2000). However, because RankSVM does not make any improvement in the ensemble stage as described in the Section 6, it is not used in generating our final results.

5.1 Random Forests

Random Forests is a tree-based learning method introduced by Breiman (2001). The algorithm constructs multiple decision trees using randomly sub-sampled features and instances. For prediction, the output is by averaging the results of individual trees. The use of multiple trees reduces the variance of prediction, so Random Forests is robust and useful in this competition.

We use the implementation in the scikit-learn package (Pedregosa et al., 2011). The package provides a parallel module to significantly speed up the tree building process. Note that the scikit-learn implementation combines classifiers by averaging probabilistic predictions instead of a voting mechanism in Breiman (2001). To construct each tree in the forest, the same number of training samples as in the original training set are sampled with replacement. Thus, the expected number of training instances for each tree is $1 - \frac{1}{e}$ times the original training set size, while some instances sampled more than once have higher weights.

In this competition, the variance may influence the standing on the leaderboard significantly. For example, with different random seeds and fewer trees, the performance of Random Forests can vibrate from 0.981 to 0.985 on the *Valid* set. On the public leaderboard, the scores of top 20 places are from 0.98130 to 0.98554. Moreover, the improvement on the *Valid* set by changing the random seed may not be consistent with the result on the self-split internal validation set. Therefore, changing the random seeds may cause over-fitting. Our experiments show that using more trees leads to better and consistent validation scores on both *Valid* set and the internal validation set due to lower variance. Because of the time limit, we use a subset of 55 features,[4] $12,000$ trees and a fixed random seed 1 in our Random Forests model after some trials. In addition to the number of trees, we also tune the minimal number of training samples in a leaf of each decision tree. This setting achieves 0.983340 MAP score on the *Valid* set. The parameters we have used are listed in Table 5.1. For unlisted parameters, we use the default values in Pedregosa et al. (2011).

5.2 Gradient Boosting Decision Tree

Gradient Boosting Decision Tree (GBDT) (Friedman, 2002), also called MART, is a tree-based learning algorithm. The goal of GBDT is using $(y, \mathbf{x})$, where $\mathbf{x}$ is the known feature vector and y is the corresponding label, to find a classifier $H^*(\mathbf{x})$ to minimize the expected

4. Because some features take more time for generation and debugging, we only use 55 stable ones to train the final Random Forests model.

Parameter	Value
Number of trees	12,000
Minimal number of samples in a leaf	10

Table 3: Tuned parameters for Random Forests.

value of the given error function $\text{err}(y, H(\mathbf{x}))$. Therefore, the target classifier is defined as

$$H^*(\mathbf{x}) = \arg\min_{H(x)} E_{y,\mathbf{x}}[\text{err}(y, H(\mathbf{x}))].$$

From the functional gradient descent perspective (Friedman, 2002), we could approximate $H^*(\mathbf{x})$ by combining several "weak" classifiers $h_t(\mathbf{x})$ as follows,

$$H_T(\mathbf{x}) = \sum_{t=0}^{T} \alpha_t h_t(\mathbf{x}),$$

where $T+1$ is the number of weak classifiers. Then we can boost the performance in an iterative manner. After we train an initial classifier h_0, for each iteration t, where $t \geq 1$, we solve the following optimization problem,

$$(h_t(\mathbf{x}), \alpha_t) = \arg\min_{h(\mathbf{x}),\alpha} \sum_{i=1}^{N} \text{err}(y_i, H_{t-1}(\mathbf{x}_i) + \alpha h(\mathbf{x}_i)),$$

where α_t is a scalar and N is the number of training instances. Then the update rule is $H_t(\mathbf{x}) = H_{t-1}(\mathbf{x}) + \alpha_t h_t(\mathbf{x})$. The GBDT is one variant of the functional gradient descent algorithm. The base (weak) classifier used in GBDT is the regression tree with constant predictions; that is, for each leaf node L, the prediction is $\frac{1}{|L|}\sum_{(\mathbf{x}_i,y_i)\in L} y_i$. To avoid overfitting, we usually use a learning rate η to shrink the effect of α_t. Therefore, the update rule becomes $H_t(\mathbf{x}) = H_{t-1}(\mathbf{x}) + \eta\alpha_t h_t(\mathbf{x})$.

Compared with Random Forests, a GBDT model is built sequentially and it combines built trees to generate a powerful learner by an iterative boosting way under the functional gradient descent perspective. We use the same package scikit-learn (Pedregosa et al., 2011). The error function of GBDT implemented in Pedregosa et al. (2011) is to optimize "deviance" which is same as the objective of logistic regression. The main disadvantage of GBDT is that it cannot be trained in parallel, so we only use 300 trees to build the final ensemble model of GBDT. This is much smaller than 12,000 for Random Forests. The tuned parameters are listed in Table 4 while the unlisted parameters are set to the default values. With the tuned parameters, the GBDT model could achieve 0.983046 MAP score on the *Valid* set.

5.3 LambdaMart

We choose LambdaMART (Wu et al., 2010) because of its recent success on Yahoo! Learning to Rank Challenge (Chapelle and Chang, 2011). LambdaMART is the combination of GBDT (Friedman, 2002) and LambdaRank (Burges et al., 2006). Burges et al. (2006) propose to use a utility function whose derivative is the gradient of a typical pairwise

Parameter	Value
Number of trees	300
Learning Rate	0.08
Tree Depth	5
Minimal number of samples in a leaf	9

Table 4: Tuned parameters for Gradient Boosting Decision Tree.

Parameter	Value
Number of trees	1,000
Minimal sample ratio in a leaf	0.01
Number of leaves	32
Ratio of sampled instances	0.3

Table 5: Tuned parameters for LambdaMART.

error function times the difference of the desired evaluation criterion, such as NDCG, by exchanging the ranking order of a pair (i, j). In contrast, GBDT (MART) aims to model the gradient in each iteration. Therefore, the main advantage of LambdaMART is that it uses LambdaRank gradients of the proposed utility function in GBDT to consider highly non-smooth ranking metrics. We use the implementation in the JForests (Ganjisaffar et al., 2011), which optimizes the NDCG metric. The detailed parameters are listed in Table 5. Compared with Random Forests and Gradient Boosting Decision Tree, LambdaMART is a more aggressive ranking algorithm. To avoid over-fitting, we train 10 LambdaMART models with random seeds from 0 to 9, and average the output confidence scores. With the listed parameters and the bagging approach, the LambdaMART model could achieve 0.983047 MAP score on the *Valid* set.

5.4 RankSVM

Besides the above tree-based models, we also explore the commonly-used RankSVM (Herbrich et al., 2000), which is extended from standard support vector machines (Vapnik, 1998). Given the author a and two papers p_i and p_j, RankSVM aims to predict p_i with a higher score than p_j, if p_i is written by the author a while p_j is not. By defining a set of pairs of the author a as

$$\mathcal{P}_a \equiv \{(p_i, p_j) \mid p_i \text{ is written by } a \text{ while } p_j \text{ is not}\}.$$

We consider the following L1-loss SVM,

$$\min_{\mathbf{w}} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C \sum_{a \in \mathcal{A}} \sum_{(i,j) \in \mathcal{P}_a} \max(0, 1 - \mathbf{w}^T(\mathbf{x}_i - \mathbf{x}_j)),$$

where $\frac{1}{2}\mathbf{w}^T\mathbf{w}$ is the regularization term and C is the regularization parameter. Due to the efficiency issue, we only study linear rather than kernel RankSVM. We consider the implementation in Lee and Lin (2014), which optimizes the L2-loss instead. The best parameter in our study is $C = 0.001$, which results in 0.97911 MAP score on the Valid set.

5.5 Summary

We summarize the results of the four studied algorithms in Table 6.

Algorithm	MAP Score
Random Forests	0.983340
Gradient Boosting Decision Tree	0.983046
LambdaMart	0.983047
RankSVM	0.979110

Table 6: The results of four studied algorithms on the *Valid* set (public leaderboard).

6. Ensemble and Post-Processing

To further boost our performance, we ensemble results of different models and conduct a post-processing procedure.

6.1 Ensemble

In many past competitions, such as Netflix and KDD Cup, winners have shown that an ensemble of individual models may significantly improve the prediction results (Tösscher et al., 2009; Yu et al., 2010; Wu et al., 2012). The main reason is that the diversification of models compensates the weakness of each model. Existing approaches to ensemble classifiers are based on some optimization techniques (Burges et al., 2005) because they often aim to combine a large number of models.

In our system, we calculate a simple weighted average after scaling the decision values of each model to be between 0 and 1. Because only four models described in Section 5 were built, we search a grid of weights to find the best setting rather than applying more complicated optimization techniques.

To see the performance under a setting of weights, we check the results on the internal validation set and the official *Valid* set. Specifically, we train four models on the internal training set, and predict on the internal validation set. Then we combine the results by adjusting weights to seek for improvements. Similarly, we train four models on the *Train* set (internal training set + internal validation set) and predict on the *Valid* set. Then we check whether results are further improved. The final weights are 0 for RankSVM (unused), 1 for both Gradient Boosting Decision Tree and LambdaMART, and 5 for Random Forests.

Based on the MAP scores reported in Section 5 and the weights for ensemble, tree-based models are more effective than the linear model in this task. This situation is similar to some ranking tasks discussed in Chapelle and Chang (2011).

6.2 Post-Processing

In post-processing stage, we remove the duplicates in the data to boost the performance. The duplicates include the paper-author pairs and paper ids.

6.2.1 Duplicated Paper-Author Pairs

In Section 4.4, we describe the concept of Heterogeneous Bibliographic Network. Even if there is an edge between the author node a and the paper node p, a may not be the author of p because of the incorrect information in `PaperAuthor.csv`. To get confidence on each link, we observe from `PaperAuthor.csv` that there are some duplicated paper-author pairs. For example, lines 147,035 and 147,036 record the same author-paper pair. We observe that duplicates highly correlate with the confirmation. Therefore, we let the number of duplicates be the weight of the edge between a paper and an author. We use weighted edges in two ways. First, we add a feature to illustrate the number of duplicates before the training procedure to obtain models described in Section 5. Second, according to the number of duplicates, we divide the given papers of each author into two groups: those having more than one duplicate and those having only one. Then in our prediction, we rank the first group before the second. For each group, we rank its members according to their decision values.

6.2.2 Duplicated Paper ID

In the *Test* set, the assigned papers of an author may contain duplicates. For example, author 100 has five papers 1, 2, 2, 3 and 4 to be ranked, and confirmed papers are 1, 2, 2 and 4. According to the algorithm provided by the competition organizer for calculating MAP, only one of these duplicated paper IDs will be calculated in MAP. Therefore, the list 1, 2, 4, 3, 2 has a higher MAP than the list 1, 2, 2, 4, 3 because the second paper with ID 2 is treated as a deleted paper in the evaluation algorithm. Based on this observation, we put all duplicated paper IDs to the end of the ranked list as deleted papers.

7. Discussion and Conclusion

In this section, we discuss some issues related to our approach and/or the KDD Cup competition. We investigate the feature importance reported by Random Forests in Table 7 because of its best performance among all the single models we used. The two most important features are related to the number of duplicates, which justifies the validity of post-processing in Section 6.2. The next two are about the affiliation consistency. Their high ranks support our observation that some mis-assignments are caused by similar names in different institutes. The features ranked next are about the name similarity between the target author and co-authors with different affiliations. Note that for these features, first name and last name are not exchanged. These features are also related to name ambiguity. If the name of the target author is the same or almost the same as a co-author of the same paper, usually the assignment is wrong.

We discuss some potential issues and difficulties for applying our method to Microsoft Academic Search or any other real online systems in practice. One potential drawback of our method in terms of scalability is the feature generation step, which may have superlinear time complexity. In particular, several coauthor name-matching features require computing the string distances between the target author and all coauthors, and each author may have several different names depending on the number of publications the author has. The computation time will be a serious issue when an author has many publications, and a

Rank	Feature	Average Importance	Standard Deviation	Rank	Feature	Average Importance	Standard Deviation
1	A.3.4	0.143027	0.003299	29	A.1.2.5	0.004102	0.000057
2	A.3.28	0.124315	0.001593	30	A.1.2.15	0.003954	0.000051
3	A.1.1.4	0.10853	0.002038	31	A.1.2.16	0.003909	0.00003
4	A.1.1.2	0.096152	0.001749	32	A.3.3	0.003824	0.000104
5	A.1.2.12	0.077095	0.000913	33	A.3.8	0.00374	0.000008
6	A.1.2.6	0.072966	0.00107	34	A.1.2.19	0.003506	0.000105
7	A.1.2.17	0.051346	0.001337	35	A.1.2.20	0.003265	0.000115
8	A.1.2.7	0.040475	0.000919	36	A.1.2.21	0.002957	0.000112
9	A.1.2.13	0.031379	0.000694	37	A.1.3.1	0.002922	0.00003
10	A.1.2.23	0.02523	0.000957	38	A.2.18	0.002588	0.000003
11	A.1.2.3	0.020658	0.000323	39	A.1.2.9	0.002302	0.000121
12	A.1.2.24	0.020075	0.000416	40	A.3.5	0.002172	0.000005
13	A.1.3.4	0.017408	0.000343	41	A.1.3.2	0.001948	0.000008
14	A.1.3.5	0.014549	0.000255	42	A.1.2.18	0.0019	0.00003
15	A.1.3.3	0.012566	0.000331	43	A.1.3.11	0.001681	0.000003
16	A.1.2.4	0.012349	0.000673	44	A.1.3.8	0.001638	0.000024
17	A.1.3.7	0.011437	0.0003	45	A.1.3.10	0.001531	0.000033
18	A.3.2	0.009053	0.000096	46	A.1.3.9	0.001498	0.00004
19	A.1.2.22	0.008349	0.000331	47	A.1.2.11	0.001468	0.000005
20	A.3.1	0.006641	0.000031	48	A.1.3.12	0.001289	0.000001
21	A.3.27	0.006436	0.000095	49	A.3.29	0.000965	0.000084
22	A.3.26	0.006392	0.000127	50	A.1.1.5	0.000917	0.00003
23	A.3.25	0.00527	0.000022	51	A.1.3.13	0.000789	0.000003
24	A.1.1.3	0.00514	0.00009	52	A.1.2.8	0.000284	0.000006
25	A.1.2.14	0.004899	0.000062	53	A.3.6	0	0
26	A.1.3.6	0.004572	0.000085	54	A.3.12	0	0
27	A.3.7	0.004355	0.000043	55	A.3.11	0	0
28	A.1.2.10	0.004185	0.000005				

Table 7: Mean and standard deviation of feature importance by training Random Forests with ten different random seeds.

paper has many authors. This situation is very common in fields such as high-energy physics. Note that feature computation is also an important issue in the prediction stage because a real-time response for the system is required. Another drawback of our method is that it cannot be updated in an incremental manner. Instead, whenever the data set is updated, features must be recomputed and the model must be retrained. To adapt the proposed system for real applications, acceleration for feature computation such as using an indexing structure (Jin et al., 2005) or conducting name grouping (Cohen et al., 2003) is necessary.

Another issue for our system (and maybe systems of other teams in this competition) is the cost effectiveness. While we use 97 different features in our final system to achieve 0.98259 MAP, we can achieve around 0.94 MAP by using one single name-matching (string distance) feature. The 0.04 MAP gain comes at a high cost in both training and testing, but whether this gain enhances users' satisfaction remains to be further investigated.

The last issue is about the data. We discussed in Section 4 that some duplicates author-paper pairs and duplicated IDs appear in the data. Although the features considering duplicates are useful in the competition, they might not be effective in practice. The cause of duplicates may be because that the system crawls data from different sources without conducting any data cleaning. If this hypothesis holds, then the number of duplicates can represent certain confidence supported by different sources and our approaches might still be valid and useful in practice. If it does not, the cause of duplicates and the usefulness of the proposed approaches remain to be further studied.

We also discuss the potential future work of our approaches. In Section 4.2.3, we assume the coauthor-affiliation consistency and research-topic consistency. In practice, it is common that an author works on more than one research topic and co-works with different institutes. Further, the affiliations and research topics of an author may change along with time. Therefore, how to model different research topics and time information into features is a topic worth studying.

In conclusion, we introduce the approaches of team National Taiwan University for track 1 of KDD Cup 2013. We successfully transform the given text information into several useful features and propose techniques to address the issue of noisy texts for making features robust. We then apply several state-of-the-art algorithms on the generated features. To further improve the performance, we conduct a simple weighted-average ensemble and a post-processing procedure by utilizing some duplicated information. During each stage, we cautiously use the internal validation or the official *Valid* set to potentially avoid the over-fitting issue. This step is crucial for us to get the best performance on the private leaderboard for predicting data in the *Test* set. A detailed summary of our approach is in Figure 2.

Acknowledgments

We thank the organizers for holding this interesting competition. We also thank the College of Electrical Engineering and Computer Science as well as the Department of Computer Science and Information Engineering at National Taiwan University for their supports and for providing a stimulating research environment. The work was also supported by National

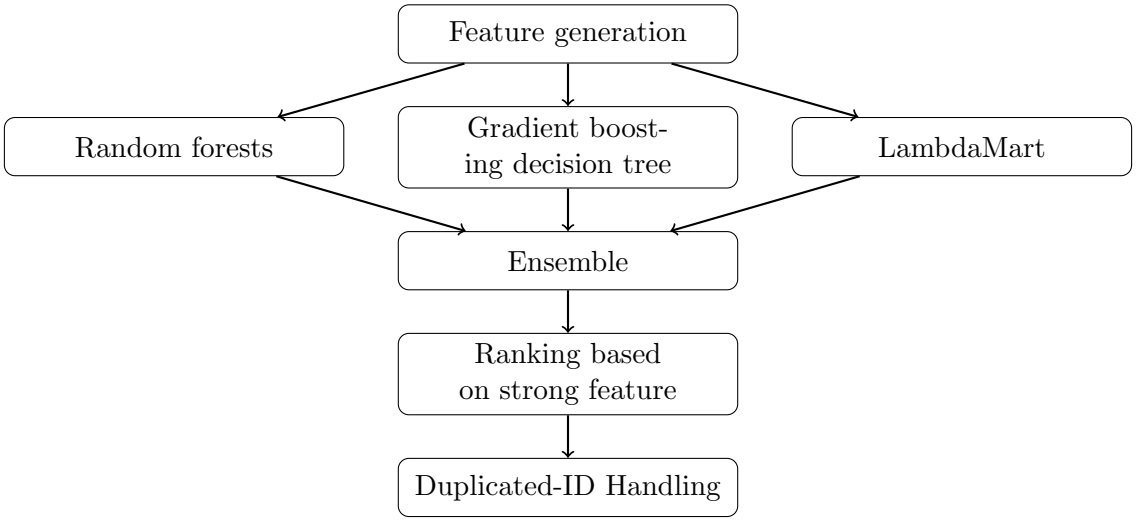


Figure 2: The detailed architecture of our approach.

Taiwan University under Grants NTU 102R7827, 102R7828, 102R7829, and by National Science Council under Grants NSC 101-2221-E002-199-MY3, 101-2628-E002-028-MY2, 101-2628-E002-029-MY2.

Appendix A. Feature List

Since our team members are divided into several sub-groups internally, some features are repeatedly generated. For these features, we denote the n times repeats by (*n) at the end of the description.

A.1 Features Using Author Information

A.1.1 Confirmation of Author Profile

1. The Levenshtein distance between the names of the target author in `Author.csv` and `PaperAuthor.csv`.

2. The Levenshtein distance between the affiliations of the target author in `Author.csv` and `PaperAuthor.csv` (*2).

3. The ratio of matched substring between the names of the target author in `Author.csv` and `PaperAuthor.csv`.

4. The ratio of matched substring between the affiliations of the target author in `Author.csv` and `PaperAuthor.csv`.

5. The ratio of matched substring between the abbreviated names of the target author in `Author.csv` and `PaperAuthor.csv`.

A.1.2 Coauthor Name Matching

1. The maximum Jaro distances between the target author's name and each coauthor's name. The names are from `PaperAuthor.csv` under the target paper.

2. The maximum Jaro distances between the last names of the target author and each coauthor. The names are from `PaperAuthor.csv` under the target paper.

3. The maximum Jaro distances between the target author's name and each coauthor's name. The names are from `PaperAuthor.csv` under the target paper. Coauthors having the same affiliation with the target author are ignored during the comparison.

4. The minimum Levenshtein distances between the target author's name and each coauthor's name. The names are from `PaperAuthor.csv` under the target paper. Coauthors that are in the same affiliation of the target author are ignored during comparison.

5. The number of authors having the same name as the target author in the entire data set.

6. The maximum Jaro distances between the abbreviated names of the target author and each coauthor. The names are from `PaperAuthor.csv` under the target paper. Coauthors that are in the same affiliation of target author are ignored during comparison.

7. The minimum among Levenshtein distances between the abbreviated names of the target author and each coauthor. The names are from `PaperAuthor.csv` under the target paper. Coauthors that are in the same affiliation of target author are ignored during comparison.

8. The minimum substring matched ratios between the target author's last name and each coauthor's last name. The author's name is form `Author.csv`, and coauthors' names are from `PaperAuthor.csv` under the target paper. Coauthors that are in the same affiliation of target author are ignored during comparison.

9. The minimum substring matched ratios between the target author's first name and each coauthor's first name. The author's name is form `Author.csv`, and coauthors' names are from `PaperAuthor.csv` under the target paper. Coauthors that are in the same affiliation of target author are ignored during comparison.

10. The minimum substring matched ratios between the target author's reversed name and each coauthor's name. Middle name is ignored, and the target author's first name and last name are exchanged before comparison. The author's name is form `Author.csv`, and coauthors' names are from `PaperAuthor.csv` under the target paper. Coauthors that are in the same affiliation of target author are ignored during comparison.

11. The minimum substring matched ratios between the target author's middle name and each coauthor's middle name. The author's name is form `Author.csv`, and coauthors' names are from `PaperAuthor.csv` under the target paper. Coauthors that are in the same affiliation of target author are ignored during comparison.

12. The maximum Jaro distances between the target author's last name and each coauthor's last name. The names are from `PaperAuthor.csv` under the target paper. Coauthors in the same affiliation as the target author are ignored during comparison.

13. The maximum Jaro distances between the target author's first name and each coauthor's first name. The names are from `PaperAuthor.csv` under the target paper. Coauthors in the same affiliation as the target author are ignored during comparison.

14. The maximum Jaro distances between the target author's name and each coauthor's name. Middle name is ignored, and the target author's first name and last name are exchanged before comparison. The names are from `PaperAuthor.csv` under the target paper. Coauthors in the same affiliation as the target author are ignored during comparison.

15. The maximum Jaro distances between the abbreviated names of the target author and each coauthor. Middle name is ignored, and the target author's first name and last name are exchanged before abbreviation. The names are from `PaperAuthor.csv` under the target paper. Coauthors in the same affiliation as the target author are ignored during comparison.

16. The maximum Jaro distances between the abbreviated names of the target author and each coauthor. Middle name is ignored, and the coauthor's first name and last name

are exchanged before abbreviation. The names are from `PaperAuthor.csv` under the target paper. Coauthors in the same affiliation as the target author are ignored during comparison.

17. The minimum Levenshtein distances between the target author's last name and each coauthor's last name. The names are from `PaperAuthor.csv` under the target paper. Coauthors in the same affiliation as the target author are ignored during comparison.

18. The minimum Levenshtein distances between the target author's first name and each coauthor's first name. The names are from `PaperAuthor.csv` under the target paper. Coauthors in the same affiliation as the target author are ignored during comparison.

19. The minimum Levenshtein distances between the target author's name and each coauthor's name. Middle name is ignored, and the target author's first name and last name are exchanged before comparison. The names are from `PaperAuthor.csv` under the target paper. Coauthors in the same affiliation as the target author are ignored during comparison.

20. The minimum Levenshtein distances between the abbreviated names of the target author and each coauthor. Middle name is ignored, and the target author's first name and last name are exchanged before abbreviation. The names are from `PaperAuthor.csv` under the target paper. Coauthors in the same affiliation as the target author are ignored during comparison.

21. The minimum Levenshtein distances between the abbreviated names of the target author and each coauthor. Middle name is ignored, and the coauthor's first name and last name are exchanged before abbreviation. The names are from `PaperAuthor.csv` under the target paper. Coauthors in the same affiliation as the target author are ignored during comparison.

22. The maximum of affiliation Jaro distances times name Levenshtein distances between target author and coauthors. Both author name and affiliation are from `PaperAuthor.csv`.

23. The maximum Jaro distances between the target author's name and each coauthor's name. The name of target author is from `Author.csv`, and that of coauthors are from `PaperAuthor.csv` under the target paper. Coauthors that are in the same affiliation of target author are ignored during comparison.

24. The minimum Levenshtein distances between the target author's name and each coauthor's name. The name of target author is from `Author.csv`, and that of coauthors are from `PaperAuthor.csv` under the target paper. Coauthors that are in the same affiliation of target author are ignored during comparison.

A.1.3 **Author Consistency**

1. The maximum Jaro distance between the affiliation of the target author and affiliations of coauthors in the paper. The affiliations are from `Author.csv`.

2. The maximum Levenshtein distance between the affiliation of the target author and affiliations of coauthors in the paper. The affiliations are from `Author.csv`.

3. The maximum Jaro distance between the affiliation of the target author and affiliations of coauthors in the paper. The affiliations are from `PaperAuthor.csv` under the target paper.

4. The minimum Levenshtein distance between the affiliation of the target author and affiliations of coauthors in the paper. The affiliations are from `PaperAuthor.csv` under the target paper.

5. The maximum Jaro distance between the affiliation of the target author and affiliations of coauthors in the paper. The affiliations are from `PaperAuthor.csv` under all papers published by a given author.

6. The maximum Levenshtein distance between the affiliation of the target author and affiliations of coauthors in the paper. The affiliations are from `PaperAuthor.csv` under all papers published by a given author.

7. The maximum Jaccard distance between the affiliation of the target author and affiliations of coauthors in the paper. The affiliations are from `PaperAuthor.csv` under all papers published by a given author.

8. The number of coauthors in the same affiliation as the target author. The affiliations are from `PaperAuthor.csv` under the target paper.

9. The number of authors with no affiliation information in `PaperAuthor.csv` under the target paper.

10. The percentage of authors with no affiliation information in `PaperAuthor.csv` under the target paper.

11. Maximum paper title Jaro distance of the target paper and papers written by the author.

12. Minimum paper title Levenshtein distance of the target paper and papers written by the author.

13. Maximum keywords Jaccard distance of the target paper and papers written by the author.

A.2 Features Using Publication Time

1. Earliest publication year of the author (*2).

2. Latest publication year of the author (*3).

3. Publication year of the paper, and the invalid year is replaced by 0 (*3).

4. Indicator to see if the publication year of the paper is missing.

5. Publication year after filling missing value.

6. Mean publication year of all papers of the author.
7. Standard deviation of publication year of all papers of the author.
8. Mean publication year of the authors' papers in the same conference as the target paper.
9. Standard deviation of the publication year of the authors' papers in the same conference as the target paper.
10. Mean publication year of the authors' papers in the same journal as the target paper.
11. Standard deviation of the publication year of the authors' papers in the same journal as the target paper.
12. Mean publication year of all papers in the same conference as the target paper.
13. Standard deviation of the publication year of all papers in the same conference as the target paper.
14. Mean publication year of all papers in the same journal as the target paper.
15. Standard deviation of the publication year of all papers in the same journal as the target paper.
16. The difference between target author's the latest publication year and the earliest publication year.
17. The difference between the target paper's publication year and the median of the publication year of all the papers of the target author.
18. The maximum publication-year difference between the target paper and papers of the target author.

A.3 Features Using Heterogeneous Bibliographic Network

1. Total number of papers published by the target author (*3).
2. Total number of coauthors of the target author (*4).
3. Number of authors of the target paper (*3).
4. Number of occurrences of the (PID,AID) pairs in `PaperAuthor.csv` (*2, and used for post processing).
5. Number of papers the author published in the conference of the target paper (*3).
6. Number of papers the author published in the journal of the target paper (*3).
7. Number of conference papers of the author (*2).
8. Percentage of conference papers of the author.
9. Number of conferences the author has papers in.

10. Number of journal papers of the author (*2).
11. Percentage of journal papers of the author.
12. Number of journals the author has papers in.
13. Average paper number of the author in conferences he/she has published in.
14. Average paper number of the author in journals he/she has published in.
15. Total number of papers written by coauthors of the target author.
16. Average paper number of coauthors of the target author.
17. The variance of paper number of coauthors of the target author.
18. Total number of papers written by coauthors in the target paper.
19. Average paper number of coauthors in the target paper.
20. The variance of paper number of coauthors in the target paper.
21. Indicator of journal papers.
22. Indicator of conference papers.
23. The difference between the number of conference papers and journal papers written by the target author.
24. The number of coauthors in the paper that have coauthored other papers with the target author.
25. The percentage of papers that are coauthored with at least one of the coauthors of the target paper.
26. Maximum number of coauthored papers with coauthors of the target paper.
27. Maximum percentage of coauthored papers (with respect to total number of papers written by the target author) with coauthors of the target paper.
28. Number of coauthors that appear more than once under the target paper in `PaperAuthor.csv`.
29. Indicator of whether the paper has only one author.
30. Number of papers published by the author which has duplicated (PID, AID) in `PaperAuthor.csv`.
31. Number of coauthored papers of the target author with all the coauthors of the target paper.
32. Number of coauthored papers of the target author with all the coauthors of the target paper, divided by the total number of coauthored papers of the target author with each coauthor of the target paper.

33. Number of coauthored papers of the target author with all the coauthors of the target paper (excluding the target paper).

34. Number of coauthored papers of the target author with all the coauthors of the target paper, divided by total number of coauthored papers of the target author with all coauthors of the target paper (excluding the target paper).

35. Total number of coauthored papers of the target author with all possible coauthors (*2).

36. Average number of coauthored papers of the target author with each coauthor of the target paper (*2).

37. Number of coauthored papers of the target author with all the coauthors of the target paper.

References

Steven Bird, Ewan Klein, and Edward Loper. Natural language processing with Python, 2009.

Leo Breiman. Random forests. *Machine Learning*, 2001.

Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd International Conference on Machine Learning*, 2005.

Christopher J. C. Burges, Robert Ragno, and Quoc Viet Le. Learning to rank with nonsmooth cost functions. In *Advances in Neural Information Processing Systems 19*, 2006.

Olivier Chapelle and Yi Chang. Yahoo! learning to rank challenge overview. *Journal of Machine Learning Research - Proceedings Track*, 2011.

William W. Cohen, Pradeep Ravikumar, and Stephen E. Fienberg. A comparison of string distance metrics for name-matching tasks. In *KDD Workshop on Data Cleaning and Object Consolidation*, pages 73–78, 2003.

Jerome H. Friedman. Stochastic gradient boosting. *Computational Statistics and Data Analysis*, 2002.

Yasser Ganjisaffar, Rich Caruana, and Cristina Lopes. Bagging gradient-boosted trees for high precision, low variance ranking models. In *Proceedings of the 34th International Conference on Research and Development in Information Retrieval*, 2011.

Ralf Herbrich, Thore Graepel, and Klaus Obermayer. *Large margin rank boundaries for ordinal regression.* MIT Press, Cambridge, MA, 2000.

Paul Jaccard. Distribution de la flore alpine dans le bassin des Dranses et dans quelques régions voisines. *Bulletin de la Société Vaudoise des Sciences Naturelles*, 1901a.

Paul Jaccard. Étude comparative de la distribution florale dans une portion des Alpes et des Jura. *Bulletin del la Société Vaudoise des Sciences Naturelles*, 1901b.

Matthew A. Jaro. Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida. *Journal of the American Statistical Association*, 1989.

Matthew A. Jaro. Probabilistic linkage of large public health data file. In *Statistics in Medicine*, 1995.

Liang Jin, Chen Li, Nick Koudas, and Anthony K. H. Tung. Indexing mixed types for approximate retrieval. In *Proceedings of the 31st International Conference on Very Large Data Bases*, pages 793–804, 2005.

Patrick Juola. Authorship attribution. *Foundations and Trends in Information Retrieval*, 2006.

Tsung-Ting Kuo, Rui Yan, Yu-Yang Huang, Perng-Hwa Kung, and Shou-De Lin. Unsupervised link prediction using aggregative statistics on heterogeneous social networks. In *Proceedings of the 19th International Conference on Knowledge Discovery and Data Mining*, 2013.

Ching-Pei Lee and Chih-Jen Lin. Large-scale linear rankSVM. *Neural Computation*, 2014. URL `http://www.csie.ntu.edu.tw/~cjlin/papers/ranksvm/ranksvml2.pdf`. To appear.

John Boaz Lee and Henry Adorna. Link prediction in a modified heterogeneous bibliographic network. In *Advances in Social Networks Analysis and Mining*, 2012.

VI Levenshtein. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 1966.

Chun-Liang Li, Yu-Chuan Su, Ting-Wei Lin, Cheng-Hao Tsai, Wei-Cheng Chang, Kuan-Hao Huang, Tzu-Ming Kuo, Shan-Wei Lin, Young-San Lin, Yu-Chen Lu, Chun-Pai Yang, Cheng-Xia Chang, Wei-Sheng Chin, Yu-Chin Juan, Hsiao-Yu Tung, Jui-Pin Wang, Cheng-Kuang Wei, Felix Wu, Tu-Chun Yin, Tong Yu, Yong Zhuang, Shou-de Lin, Hsuan-Tien Lin, and Chih-Jen Lin. Combination of feature engineering and ranking models for paper-author identification in kdd cup 2013. In *KDD Cup 2013 Workshop*, 2013.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 2011.

Senjuti Basu Roy, Martine De Cock, Vani Mandava, Swapna Savanna, Brian Dalessandro, Claudia Perlich, William Cukierski, and Ben Hamner. The microsoft academic search dataset and kdd cup 2013. In *KDD Cup 2013 Workshop*, 2013.

Efstathios Stamatatos. A survey of modern authorship attribution methods. *Journal of the American Society for Information Science and Technology*, 2009.

Yizhou Sun, Rick Barber, Manish Gupta, Charu C Aggarwal, and Jiawei Han. Co-author relationship prediction in heterogeneous bibliographic networks. In *Advances in Social Networks Analysis and Mining*, 2011.

Yizhou Sun, Jiawei Han, Charu C. Aggarwal, and Nitesh V. Chawla. When will it happen?: Relationship prediction in heterogeneous information networks. In *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*, 2012.

Andreas Tösscher, Michael Jahrer, and Robert M. Bell. The bigchaos solution to the netflix grand prize. Technical report, 2009.

Vladimir N. Vapnik. *Statistical learning theory.* Wiley, 1998.

Kuan-Wei Wu, Chun-Sung Ferng, Chia-Hua Ho, An-Chun Liang, Chun-Heng Huang, Wei-Yuan Shen, Jyun-Yu Jiang, Ming-Hao Yang, Ting-Wei Lin, Ching-Pei Lee, Perng-Hwa Kung, Chin-En Wang, Ting-Wei Ku, Chun-Yen Ho, Yi-Shu Tai, I-Kuei Chen, Wei-Lun Huang, Che-Ping Chou, Tse-Ju Lin, Han-Jay Yang, Yen-Kai Wang, Cheng-Te Li, Shou-De Lin, and Hsuan-Tien Lin. A two-stage ensemble of diverse models for advertisement ranking in KDD cup 2012. Technical report, Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, 2012. URL `http://www.csie.ntu.edu.tw/~htlin/paper/doc/wskdd12cup.pdf`.

Qiang Wu, Christopher J. C. Burges, Krysta Marie Svore, and Jianfeng Gao. Adapting boosting for information retrieval measures. *Information Retrieval*, 2010.

Yang Yang, Nitesh V. Chawla, Yizhou Sun, and Jiawei Han. Link prediction in heterogeneous networks: Influence and time matters. Technical report, 2012.

Hsiang-Fu Yu, Hung-Yi Lo, Hsun-Ping Hsieh, Jing-Kai Lou, Todd G. McKenzie, Jung-Wei Chou, Po-Han Chung, Chia-Hua Ho, Chun-Fu Chang, Yin-Hsuan Wei, Jui-Yu Weng, En-Syu Yan, Che-Wei Chang, Tsung-Ting Kuo, Yi-Chen Lo, Po T. Chang, Chieh Po, Chien-Yuan Wang, Yi-Hung Huang, Chen-Wei Hung, Yu-Xun Ruan, Yu-Shi Lin, Shou-De Lin, Hsuan-Tien Lin, and Chih-Jen Lin. Feature engineering and classifier ensemble for KDD cup 2010. Technical report, Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, 2010. URL `http://www.csie.ntu.edu.tw/~cjlin/courses/dmcase2010/kdd2010ntu.pdf`.

Xiao Yu, Quanquan Gu, Mianwei Zhou, and Jiawei Han. Citation prediction in heterogeneous bibliographic networks. In *SIAM International Conference on Data Mining*, 2012.

Journal of Machine Learning Research 16 (2015) 2949-2997 Submitted 10/12; Revised 2/15; Published 12/15

Comparing Hard and Overlapping Clusterings

Danilo Horta HORTA@ICMC.USP.BR
Ricardo J. G. B. Campello CAMPELLO@ICMC.USP.BR
Instituto de Ciências Matemáticas e de Computação
Universidade de São Paulo – Campus de São Carlos
Caixa Postal 668, 13560-970, São Carlos-SP, Brazil

Editor: Marina Meila

Abstract

Similarity measures for comparing clusterings is an important component, e.g., of evaluating clustering algorithms, for consensus clustering, and for clustering stability assessment. These measures have been studied for over 40 years in the domain of exclusive hard clusterings (exhaustive and mutually exclusive object sets). In the past years, the literature has proposed measures to handle more general clusterings (e.g., fuzzy/probabilistic clusterings). This paper provides an overview of these new measures and discusses their drawbacks. We ultimately develop a corrected-for-chance measure (13AGRI) capable of comparing exclusive hard, fuzzy/probabilistic, non-exclusive hard, and possibilistic clusterings. We prove that 13AGRI and the adjusted Rand index (ARI, by Hubert and Arabie) are equivalent in the exclusive hard domain. The reported experiments show that only 13AGRI could provide both a fine-grained evaluation across clusterings with different numbers of clusters and a constant evaluation between random clusterings, showing all the four desirable properties considered here. We identified a high correlation between 13AGRI applied to fuzzy clusterings and ARI applied to hard exclusive clusterings over 14 real data sets from the UCI repository, which corroborates the validity of 13AGRI fuzzy clustering evaluation. 13AGRI also showed good results as a clustering stability statistic for solutions produced by the expectation maximization algorithm for Gaussian mixture. Implementation and supplementary figures can be found at `http://sn.im/25a9h8u`.

Keywords: overlapping, fuzzy, probabilistic, clustering evaluation

1. Introduction

Clustering is a task that aims to determine a finite set of categories (clusters) to describe a data set according to similarities/dissimilarities among its objects (Kaufman and Rousseeuw, 1990; Everitt et al., 2001). Several clustering algorithms are published every year, which makes developing of effective measures to compare clusterings indispensable (Vinh et al., 2009, 2010). Clustering algorithm A is commonly considered better than B for a given data set X if A produces clusterings that are more similar (according to a similarity measure[1] for clustering) to a reference solution for X than those produced by B. Similarity measures are also used for consensus clustering, clustering stability assessment, and even for quantifying information loss (Strehl and Ghosh, 2003; Monti et al., 2003; Yu

1. Note that a dissimilarity/distance measure can always be cast into a similarity measure. For comparison purposes, we transformed dissimilarity/distance measures into similarity measures in this work.

et al., 2007; Beringer and Hllermeier, 2007; Vinh and Epps, 2009). A consensus clustering technique aims to find a high-quality clustering solution by combining several (potentially poor) solutions obtained from different methods, algorithm initializations, or perturbations of the same data set. This combination is achieved by producing a solution that shares the most information, quantified by a similarity measure, with the original solutions (Strehl and Ghosh, 2003). In the context of clustering stability assessment, the method used to generate a set of clustering solutions is considered stable if the set shows low variation, which is considered a desirable quality (Kuncheva and Vetrov, 2006). One can apply a clustering algorithm several times to subsamples of the original data set for any numbers of clusters, producing a set of clusterings for each number of clusters. The number of clusters for which the set of solutions is less diverse is considered a good estimate of the true number of clusters (Borgelt and Kruse, 2006; Vinh and Epps, 2009). Another interesting application of similarity measures is in the quantification of information loss (Beringer and Hllermeier, 2007). To increase efficiency (e.g., in the context of data stream clustering), one can first map the data into a low-dimensional space and cluster the transformed data. If the transformation is almost lossless, the clustering structures in the two spaces should be highly similar; a similarity measure can be used to assess this.

Several established measures are suitable for comparing exclusive hard clusterings (EHCs) (Albatineh et al., 2006; Meila, 2007; Vinh et al., 2009, 2010), i.e., clusterings in which each object exclusively belongs to one cluster. Examples of popular measures are the Rand index (RI) (Rand, 1971), adjusted Rand index (ARI) (Hubert and Arabie, 1985), Jaccard index (JI) (Jaccard, 1908), mutual information (Strehl and Ghosh, 2003), and variation of information (VI) (Meila, 2005). Bcubed (BC) (Bagga and Baldwin, 1998; Amigó et al., 2009) is a measure for evaluating coreferences (e.g., a set of pronouns referring to the same noun in a paragraph) in the natural language processing field. Coreferences can also be viewed as EHCs (Cardie and Wagstaf, 1999), and BC satisfies some (frequently regarded as) desirable properties that most well-known EHC measures do not (Amigó et al., 2009). Thus, we also include BC in this work. There are other important clustering types, e.g., fuzzy/probabilistic clustering[2] (FC), non-exclusive hard clustering (NEHC), and possibilistic clustering (PC) (Campello, 2010; Anderson et al., 2010), that are not assessed using well-established measures but that would benefit from the tasks discussed above.

Various EHC measure generalizations have recently appeared in the literature (Borgelt and Kruse, 2006; Campello, 2007; Anderson et al., 2010; Campello, 2010) to fill this gap. Unfortunately, all these measures exhibit critical problems that hinder their applicability. The RI fuzzy version by Campello (2007) does not attain its maximum (i.e., 1) whenever two identical solutions are compared, which makes it difficult to convey the similarity of the compared solutions. The same issue is exhibited by other RI generalizations (Borgelt and Kruse, 2006; Ceccarelli and Maratea, 2008; Rovetta and Masulli, 2009; Brouwer, 2009; Anderson et al., 2010; Quere and Frelicot, 2011). Moreover, most of the proposed measures are not corrected for randomness, i.e., they do not provide a constant average evaluation

2. The usage of "fuzzy" or "probabilistic" depends on the interpretation of the object membership degrees given by the solution. Fuzzy c-means (Bezdek, 1981) and expectation maximization (EM) (Dempster et al., 1977) give a fuzzy and a probabilistic interpretation, respectively, although the solutions they produce come from the same domain of clusterings. We will hereafter call it fuzzy clustering in both cases for simplicity.

over sets of independently generated clusterings (constant baseline for short). In practice this means that theses measures tend to favor clusterings with certain numbers of clusters (Vinh et al., 2009, 2010), whether the compared solutions are similar or not. Additionally, several of the measures have a low sensitivity to differences in solution quality, where close evaluation values can result from comparing very similar or very different solutions.

Biclustering is also an important type of clustering solution, which is usually represented by a set of pairs $C \triangleq \{(C_1^e, C_1^c), (C_2^e, C_2^c), \ldots, (C_k^e, C_k^c)\}$. Each pair (C_r^e, C_r^c) has two non-empty sets of objects of different types. In gene expression analysis, C_r^e could be the set of genes related to the experimental conditions in C_r^c (Madeira and Oliveira, 2004). In subspace clustering, C_r^e could be the set of objects related to the object features in C_r^c (Patrikainen and Meila, 2006; Günnemann et al., 2011). We do not consider this type of clustering henceforth as it would overly extend the length and complexity of this work. Moreover, a biclustering can always be converted to an NEHC (Patrikainen and Meila, 2006), which is one of the scenarios we investigate here.

We first develop an RI generalization, called the frand index (13FRI),[3] to handle FCs. We then develop the adjusted frand index (13AFRI) by correcting 13FRI for randomness. Although the assumed randomness model is apparently unrelated to that assumed for ARI (Hubert and Arabie, 1985), we prove that 13AFRI and ARI are different formulations of the same measure in the EHC domain. Finally, we also extend the 13FRI and 13AFRI measures to the more general domain of PCs (which include the NEHC, FC, and EHC solutions as special cases, Section 3), resulting in the grand index (13GRI) and adjusted grand index (13AGRI), respectively.

We defined four clearly desirable properties that a good similarity measure should display. Under this framework, our proposed measures are empirically compared in two experiments with 32 others, out of which 28 are measures proposed in the past recent years to handle more general clusterings than EHCs. Several of the measures could not distinguish among solutions that are close to from those that are far from the reference solution according to the number of clusters in the first experiment. 13AGRI presented an evident, desirable sensitivity over the ranges of the numbers of clusters. In the second experiment, 13AGRI was the only measure that exhibited a constant baseline for all scenarios of randomly generated exclusive hard, fuzzy, non-exclusive hard, and possibilistic clusterings.

We applied 13AGRI and ARI to evaluate fuzzy c-means (Bezdek, 1981) and k-means (MacQueen, 1967) solutions, respectively, over 14 real data sets from UCI repository (Newman and Asuncion, 2010). We argue that the high correlation found between 13AGRI and ARI evaluations is an indication of the 13AGRI evaluation appropriateness for FCs. 13AGRI is also assessed as a stability statistic for FCs produced by the expectation maximization for Gaussian mixture (EMGM) (Dempster et al., 1977) algorithm.

The remainder of the paper is organized as follows. Section 2 discusses evaluation of similarity measures and establishes four desirable properties. Section 3 sets the background of the work and reviews the measures proposed in the past years to tackle more general clusterings than EHCs. Section 4 presents the 13FRI measure for handling FCs, develops a corrected-for-chance version of 13FRI named 13AFRI, and explains why 13FRI and

3. The number 13 is a reminder of the publication year of the measure (2013). We use a reminder in front of each measure acronym, except for RI, ARI, JI, and BC. This helps us identify the recently proposed measures.

13AFRI are not suitable for comparing every type of PC. Section 5 proposes the 13GRI and 13AGRI measures by addressing the issue that prevented 13FRI and 13AFRI from being appropriately applied to PCs. Section 6 deduces the asymptotic computational complexity of 13FRI, 13AFRI, 13GRI, and 13AGRI and introduces an efficient algorithm to calculate the expectations used by 13AFRI and 13AGRI. Section 7 presents four experiments, the first two to empirically evaluate the measures according to the four desirable properties. First experiment (Section 7.1) assesses how the measures behave when comparing solutions produced by clustering algorithms with reference solutions across a range of the numbers of clusters. Second experiment (Section 7.2) assesses the ability of the measures to provide unbiased evaluations in several scenarios. Third experiment (Section 7.3) compares 13AGRI and ARI evaluations of fuzzy and exclusive hard clusterings in 14 real data sets. Fourth experiment (Section 7.4) uses 13AGRI as a stability statistic for FC assessment in five real data sets. Section 8 discusses the criteria adopted to evaluate and compare the measures. Section 9 concludes the work, and Appendix proves some properties of our measures.

2. Desirable Measure Properties

Evaluating a measure for comparing clusterings is a difficult task. Partly because different applications may require different perspectives regarding the similarity between clusterings, and partly because there is no universally accepted set of properties that a measure for comparing clusterings must have. It is often the case that a measure is modified to comply with a set of desirable properties but, as a side effect, loses another set of desirable properties that it previously had. This is the case of variation of information (Meila, 2005) and its corrected-for-chance version developed in (Vinh et al., 2009, 2010), where the latter gives away the metric property to gain the property of displaying constant baseline evaluations for randomly generated solutions. There is even a result stating that no "sensible" measure for comparing clusterings will simultaneously satisfy three desirable properties (Meila, 2005).

In order to evaluate the usefulness of our proposed measure, we compare ours with the ones found in the literature over four clearly desirable properties. These properties have been chosen because they are appealing from a practical perspective and together they can unveil flaws of several existing measures according to well established intuitions. The properties are defined as follows:

- **Maximum.** A measure is told to obey this property if it attains its known maximum value whenever two equivalent solutions are compared. The maximum has to be invariant to the data set as well.
- **Discriminant.** A good measure must be able to detect the best solution among a given set of solutions.
- **Contrast.** A good measure must provide progressively better (or worse) evaluations for progressively better (or worse) solutions.
- **Baseline.** A measure that has a predetermined expected value over randomly generated solutions is told to have the baseline property (also, adjusted for chance).

It is a common practice to have the maximum equal to 1 and the baseline value equal to 0, such that having the maximum property means that the measure attains 1 when comparing

two equivalent solutions and having the baseline property means that comparing randomly generated solutions tend to give evaluations close to zero.

A measure having a known maximum that is always attained when two equivalent solutions are compared provides an objective goal (i.e., producing a clustering that attains that score) and ensures the user that a better solution can be found when the evaluation is lower than the maximum. Comparisons between evaluations of clusterings generated from different data sets may be misguided because of different extents to which variation is possible when the measure does not have a fixed maximum (Luo et al., 2009). As mentioned by Vinh et al. (2010), the fact that all of the 22 different pair counting based measures discussed in (Albatineh et al., 2006) are normalized to have a known maximum further stresses the particular interest of the clustering community in this property.

A measure may not attain its predefined maximum for the ideal solution, but still might be able to detect the best solution among a set of non-ideal solutions. This elicits the measure as having the discriminant property. This property definition naturally prompts the question "How can I know that a given solution is better than another one?" that the measure tries to answer in the first place. However, there is one situation where the answer is unquestionable: any reasonable measure should evaluate the ideal solution (i.e., the one equivalent to the reference solution) as being superior to the others. If a measure somehow evaluates a given solution better than the reference one, it is clearly flawed as a similarity measure.

We propose the contrast property because we observed in preliminary experiments that some measures would give flat evaluations over solutions progressively farther from the reference one. This behavior can be problematic when such a measure is used for assessing clustering algorithms with similar accuracy, as the measure might not be sensible enough to capture any difference.

The contrast property is also related to the useful range of a measure (Fowlkes and Mallows, 1983; Wu et al., 2009; Vinh et al., 2010). A measure can have known upper and lower bounds but its evaluations can be spread out only over a small fraction of that range in practice. As an example, for a given number of objects n, RI attains the maximum 1 for two equivalent clusterings and the minimum 0 when comparing a clustering having one cluster and a clustering having n clusters. However, it has been reported that RI provides evaluations almost always above 0.5, even when comparing randomly generated clusterings (Fowlkes and Mallows, 1983; Wu et al., 2009). Knowing beforehand the useful range (i.e., the range within which the evaluations will fall for real applications) certainly increases the intuitiveness of the measure.

The maximum property can be mathematically proved for each measure, but the other properties can only be experimentally assessed and/or disproved. The discriminant and contrast properties are somewhat subjective, but a measure that evaluates the ideal solution worse than another solution clearly does not comply with those properties. The baseline property does not specify a particular model for randomly generating solutions (and we believe that specifying one would be artificial). We thus empirically evaluate the measures regarding this property over different models of randomly generating solutions.

U/V	$\mathrm{V}_{1,:}$	$\mathrm{V}_{2,:}$	$\cdots$	$\mathrm{V}_{k_\mathrm{V},:}$	Sums
$\mathrm{U}_{1,:}$	$\mathrm{N}_{1,1}$	$\mathrm{N}_{1,2}$	$\cdots$	$\mathrm{N}_{1,k_\mathrm{V}}$	$\mathrm{N}_{1,+}$
$\mathrm{U}_{2,:}$	$\mathrm{N}_{2,1}$	$\mathrm{N}_{2,2}$	$\cdots$	$\mathrm{N}_{2,k_\mathrm{V}}$	$\mathrm{N}_{2,+}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
$\mathrm{U}_{k_\mathrm{U},:}$	$\mathrm{N}_{k_\mathrm{U},1}$	$\mathrm{N}_{k_\mathrm{U},2}$	$\cdots$	$\mathrm{N}_{k_\mathrm{U},k_\mathrm{V}}$	$\mathrm{N}_{k_\mathrm{U},+}$
Sums	$\mathrm{N}_{+,1}$	$\mathrm{N}_{+,2}$	$\cdots$	$\mathrm{N}_{+,k_\mathrm{V}}$	$\mathrm{N}_{+,+}$

Table 1: Contingency table.

3. Background and Related Work

Let $X \triangleq \{\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_n\}$ be a data set with n objects. A clustering solution with k clusters can be represented by a matrix $\mathrm{U} \triangleq [\mathrm{U}_{r,i}] \in \mathbb{R}^{k \cdot n}$, where $\mathrm{U}_{r,i}$ expresses the membership degree of x_i to the rth cluster and U satisfies the following properties:

$$0 \leq \mathrm{U}_{r,i} \leq 1 \qquad (\forall r \in \mathbb{N}_{1,k} \text{ and } \forall i \in \mathbb{N}_{1,n}), \tag{1a}$$

$$0 < \textstyle\sum_{i=1}^{n} \mathrm{U}_{r,i} \qquad (\forall r \in \mathbb{N}_{1,k}), \text{ and} \tag{1b}$$

$$0 < \textstyle\sum_{r=1}^{k} \mathrm{U}_{r,i} \qquad (\forall i \in \mathbb{N}_{1,n}). \tag{1c}$$

We say that $\mathrm{U} \in M_\mathrm{p} \triangleq \{\mathrm{U} \in \mathbb{R}^{k \cdot n} \mid \text{satisfies Equations (1)}\}$ is a *possibilistic clustering* (PC). By adding more constraints, three other clustering types emerge: $\mathrm{U} \in M_\mathrm{f} \triangleq \{\mathrm{U} \in M_\mathrm{p} \mid \sum_{r=1}^{k} \mathrm{U}_{r,i} = 1 \quad \forall i\}$ is a *fuzzy/probabilistic clustering* (FC), $\mathrm{U} \in M_\mathrm{neh} \triangleq \{\mathrm{U} \in M_\mathrm{p} \mid \mathrm{U}_{r,i} \in \{0,1\} \quad \forall r, i\}$ is a *non-exclusive hard clustering* (NEHC), and $\mathrm{U} \in M_\mathrm{eh} \triangleq M_\mathrm{f} \cap M_\mathrm{neh}$ is an *exclusive hard clustering* (EHC) (Campello, 2010; Anderson et al., 2010). Note that $M_\mathrm{eh} \subset M_\mathrm{f}$, $M_\mathrm{eh} \subset M_\mathrm{neh}$, $M_\mathrm{f} \subset M_\mathrm{p}$, and $M_\mathrm{neh} \subset M_\mathrm{p}$ (Figure 1). Set M_p of all PCs covers the other sets, and a measure for this domain is applicable to virtually every type of clustering present in the literature.

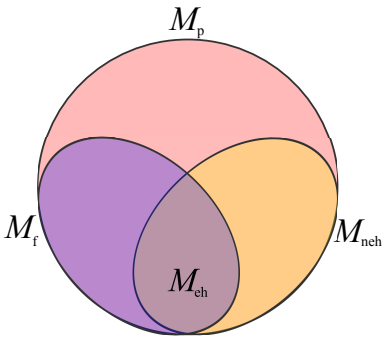


Figure 1: Venn diagram representing the relationship between clustering domains.

We believe that the most popular measures for comparing EHCs are those based on pair counting, including ARI and JI. A common approach to compute these measures begins by obtaining a contingency matrix (Albatineh et al., 2006). Let U and V be two EHCs with k_U and k_V clusters, respectively, of the same data set of n objects. Table 1 defines their contingency table, where $\mathrm{N} = \mathrm{U}\mathrm{V}^\mathrm{T}$ is the contingency matrix and $\mathrm{N}_{r,t}$ is the number

of objects that simultaneously belong to the rth cluster of U and tth cluster of V. The marginal totals $\mathrm{N}_{+,t} = \sum_{r=1}^{k_\mathrm{U}} \mathrm{N}_{r,t}$ and $\mathrm{N}_{r,+} = \sum_{t=1}^{k_\mathrm{V}} \mathrm{N}_{r,t}$ yield the cluster sizes and the grand total $\mathrm{N}_{+,+} = \sum_{r,t=1}^{k_\mathrm{U},k_\mathrm{V}} \mathrm{N}_{r,t} = n$ yields the number of objects in the data set. The contingency matrix is then used to calculate the pairing variables a (the number of object pairs in the same cluster in both U and V), b (the number of object pairs in the same cluster in U but in different clusters in V), c (the number of object pairs in different clusters in U but in the same cluster in V), and d (the number of object pairs in different clusters in both U and V) (Jain and Dubes, 1988; Albatineh et al., 2006):

$$a = \sum_{r,t=1}^{k_\mathrm{U},k_\mathrm{V}} \binom{\mathrm{N}_{r,t}}{2} = \frac{1}{2} \sum_{r,t=1}^{k_\mathrm{U},k_\mathrm{V}} \mathrm{N}_{r,t}^2 - \frac{\mathrm{N}_{+,+}}{2}, \tag{2a}$$

$$b = \sum_{r=1}^{k_\mathrm{U}} \binom{\mathrm{N}_{r,+}}{2} - a = \frac{1}{2} \sum_{r=1}^{k_\mathrm{U}} \mathrm{N}_{r,+}^2 - \frac{1}{2} \sum_{r,t=1}^{k_\mathrm{U},k_\mathrm{V}} \mathrm{N}_{r,t}^2, \tag{2b}$$

$$c = \sum_{t=1}^{k_\mathrm{V}} \binom{\mathrm{N}_{+,t}}{2} - a = \frac{1}{2} \sum_{t=1}^{k_\mathrm{V}} \mathrm{N}_{+,t}^2 - \frac{1}{2} \sum_{r,t=1}^{k_\mathrm{U},k_\mathrm{V}} \mathrm{N}_{r,t}^2, \text{ and} \tag{2c}$$

$$d = \binom{\mathrm{N}_{+,+}}{2} - (a+b+c) = \frac{1}{2}\mathrm{N}_{+,+}^2 - \frac{1}{2}\Big(\sum_{r=1}^{k_\mathrm{U}} \mathrm{N}_{r,+}^2 + \sum_{t=1}^{k_\mathrm{V}} \mathrm{N}_{+,t}^2\Big) + \frac{1}{2} \sum_{r,t=1}^{k_\mathrm{U},k_\mathrm{V}} \mathrm{N}_{r,t}^2. \tag{2d}$$

Albatineh et al. (2006) list 22 measures based on pair counting defined solely using a, b, c, and d. For example, JI and RI are respectively defined as

$$\mathrm{JI}(\mathrm{U},\mathrm{V}) \triangleq a/(a+b+c) \text{ and} \tag{3}$$

$$\mathrm{RI}(\mathrm{U},\mathrm{V}) \triangleq (a+d)/(a+b+c+d). \tag{4}$$

ARI is defined as (Hubert and Arabie, 1985)[4]

$$\mathrm{ARI}(\mathrm{U},\mathrm{V}) \triangleq \frac{a - \frac{(a+c)(a+b)}{a+b+c+d}}{\frac{(a+c)+(a+b)}{2} - \frac{(a+c)(a+b)}{a+b+c+d}}. \tag{5}$$

As an alternative to the contingency matrix, one can define the pairing variables by employing the co-association matrices $\mathrm{J}^\mathrm{U} \triangleq \mathrm{U}^\mathrm{T}\mathrm{U}$ and $\mathrm{J}^\mathrm{V} \triangleq \mathrm{V}^\mathrm{T}\mathrm{V}$ (Zhang et al., 2012). When U and V are EHCs, the above definition amounts to

$$\mathrm{J}_{i,j}^\mathrm{U} = \begin{cases} 1 & \text{if } \exists r \text{ such that } \mathrm{U}_{r,i} = 1 \text{ and } \mathrm{U}_{r,j} = 1 \\ 0 & \text{otherwise} \end{cases}. \tag{6}$$

The pairing variables can be rewritten as[5]

$$\begin{array}{ll} a = \sum_{i<j} \mathrm{J}_{i,j}^\mathrm{U}\mathrm{J}_{i,j}^\mathrm{V}, & b = \sum_{i<j} \mathrm{J}_{i,j}^\mathrm{U}(1-\mathrm{J}_{i,j}^\mathrm{V}), \\ c = \sum_{i<j}(1-\mathrm{J}_{i,j}^\mathrm{U})\mathrm{J}_{i,j}^\mathrm{V}, \text{ and} & d = \sum_{i<j}(1-\mathrm{J}_{i,j}^\mathrm{U})(1-\mathrm{J}_{i,j}^\mathrm{V}). \end{array} \tag{7}$$

4. Equation (5) in (Hubert and Arabie, 1985) for ARI is defined by combinations. However, it is equivalent to Equation (5) defined here, as $a = \sum_{r,t=1}^{k_\mathrm{U},k_\mathrm{V}} \binom{\mathrm{N}_{r,t}}{2}$, $a+b = \sum_{r=1}^{k_\mathrm{U}} \binom{\mathrm{N}_{r,+}}{2}$, and $a+c = \sum_{t=1}^{k_\mathrm{V}} \binom{\mathrm{N}_{+,t}}{2}$.

5. $\sum_{i<j}$ is a shorthand for $\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}$.

BC is based on bcubed precision (BCP) and bcubed recall (BCR) (Amigó et al., 2009):

$$\mathrm{BCP}(\mathrm{U},\mathrm{V}) \triangleq \frac{1}{n}\sum_{i=1}^{n}\frac{\sum_{j=1}^{n}\mathrm{J}^{\mathrm{U}}_{i,j}\mathrm{J}^{\mathrm{V}}_{i,j}}{\sum_{j=1}^{n}\mathrm{J}^{\mathrm{U}}_{i,j}} \quad \text{and} \tag{8a}$$

$$\mathrm{BCR}(\mathrm{U},\mathrm{V}) \triangleq \frac{1}{n}\sum_{i=1}^{n}\frac{\sum_{j=1}^{n}\mathrm{J}^{\mathrm{U}}_{i,j}\mathrm{J}^{\mathrm{V}}_{i,j}}{\sum_{j=1}^{n}\mathrm{J}^{\mathrm{V}}_{i,j}}. \tag{8b}$$

BC is defined by default as:

$$\mathrm{BC}(\mathrm{U},\mathrm{V}) \triangleq 2\cdot\frac{\mathrm{BCP}(\mathrm{U},\mathrm{V})\cdot\mathrm{BCR}(\mathrm{U},\mathrm{V})}{\mathrm{BCP}(\mathrm{U},\mathrm{V})+\mathrm{BCR}(\mathrm{U},\mathrm{V})}.$$

3.1 Similarity Measures for Clustering

Table 2 provides an overview of recently proposed measures designed to handle more general solutions than EHCs. For each measure, this table shows the clustering types for which it was designed and the approach used in its formulation.

03VI, 03MI, and 05MI are three measures based on information theory (Mackay, 2003). Let U and V be two FCs with k_{U} and k_{V} clusters, respectively. The joint probability $\mathrm{P}(r,t)$ of an object belonging to both the rth cluster in U and tth cluster in V is defined by dividing the contingency matrix N by n, i.e. $\mathrm{P}(r,t) \triangleq \mathrm{N}_{r,t}/n$. The mutual information between U and V is defined as:

$$\mathrm{I}(\mathrm{U},\mathrm{V}) \triangleq \sum_{r,t=1}^{k_{\mathrm{U}},k_{\mathrm{V}}}\mathrm{P}(r,t)\log\left(\frac{\mathrm{P}(r,t)}{\mathrm{P}(r,\cdot)\mathrm{P}(\cdot,t)}\right),$$

where $\mathrm{P}(r,\cdot) \triangleq \sum_{t=1}^{k_{\mathrm{V}}}\mathrm{P}(r,t)$ and $\mathrm{P}(\cdot,t) \triangleq \sum_{r=1}^{k_{\mathrm{U}}}\mathrm{P}(r,t)$ are the marginals. The entropy associated with U is

$$\mathrm{H}(\mathrm{U}) \triangleq \sum_{r=1}^{k_{\mathrm{U}}}\mathrm{P}(r,\cdot)\log\left(\mathrm{P}(r,\cdot)\right).$$

The 03VI, 03MI, and 05MI measures are defined as:

$$\begin{aligned}
\mathrm{03VI}(\mathrm{U},\mathrm{V}) &\triangleq \mathrm{H}(\mathrm{U})+\mathrm{H}(\mathrm{V})-2\mathrm{I}(\mathrm{U},\mathrm{V}),\\
\mathrm{03MI}(\mathrm{U},\mathrm{V}) &\triangleq \mathrm{I}(\mathrm{U},\mathrm{V})/\sqrt{\mathrm{H}(\mathrm{U})\mathrm{H}(\mathrm{V})}, \text{ and}\\
\mathrm{05MI}(\mathrm{U},\mathrm{V}) &\triangleq 2\mathrm{I}(\mathrm{U},\mathrm{V})/(\mathrm{H}(\mathrm{U})+\mathrm{H}(\mathrm{V})).
\end{aligned}$$

We assume base two for $\log(\cdot)$ in the experiments (Section 7).

07CRI was developed based on a set-theoretic formulation of pairing variables. Let U and V be two EHCs. Let R be the set of unordered object pairs belonging to the same cluster in U, and let T be the set of unordered object pairs belonging to the same cluster in V. The usual cardinality $|R\cap T|$ yields the pairing variable a; using the same approach, variables b, c, and d can be defined by their sets. Fuzzy versions of the pairing variables were then defined by replacing the usual set operations with counterparts from fuzzy set

Measure	EHC	FC	NEHC	PC	Based on
03VI (Meila, 2003) 03MI (Strehl and Ghosh, 2003) 05MI (Fred and Jain, 2005)	*	*			Information theory
07CRI (Campello, 2007) 07CARI	*	*	*	*	Fuzzy sets (a, b, c, d)
08BRIp (Borgelt, 2007) 08BRIm	*	*			J^{U} (a, b, c, d)
09EBC (Amigó et al., 2009)	*		*		Precision/Recall
09CRI (Ceccarelli and Maratea, 2009) 09CARI	*	*	*	*	$\dot{\mathrm{N}}$ (a, b, c, d)†
09HI (Hullermeier and Rifqi, 2009)	*	*	*	*	Dist. ($\mathrm{U}_{:,i}$ and $\mathrm{U}_{:,j}$)
09RI (Rovetta and Masulli, 2009)	*	*			J^{U} (ad hoc)
09BRI (Brouwer, 2009) 09BARI	*	*	*	*	J^{U} (ad hoc)
10QRIp (Quere et al., 2010) 10QRIm	*	*	*	*	J^{U} (a, b, c, d)
10ARI (Anderson et al., 2010) 10AARI 10ARIn 10AARIn	*	*	*	*	N (a, b, c, d)*
10CSI (Campello, 2010)	*		*		ad hoc
10CF (Campello, 2010) 10CFn	*	*	*	*	Edit distance
11ARInm (Anderson et al., 2011) 11AARInm	*	*	*	*	N (a, b, c, d)*
11MD (Wang, 2010)	*		*		J^{U} (ad hoc)
11D2 (Wang, 2010)	*		*		Hamming distance
12DB (Wang, 2012)	*		*		Information theory

† The contingency matrix N used is not the same as the original one. Ceccarelli and Maratea (2009) it defined as $\dot{\mathrm{N}}_{r,t} \triangleq \sum_{i=1}^{n}(\mathrm{U}_{r,i} + \mathrm{V}_{t,i})^{\alpha}$. We adopt $\alpha \triangleq 1$ for simplicity.

* Measures 10ARIn, 10AARIn, 11ARInm, and 11AARInm use a normalized contingency matrix $\hat{\mathrm{N}}$.

Table 2: General similarity measures.

theory (Campello, 2007). Plugging the new versions of a, b, c, and d into Equations (4) and (5) resulted in 07CRI and 07CARI, respectively, where U and V are PCs.

08BRIp and 08BRIm are RI generalizations based on the definitions of a, b, c, and d given by Equations (7), where an arbitrary t-norm (from fuzzy set theory) replaces the multiplication operator used to compute $\mathrm{J}^{\mathrm{U}} = \mathrm{U}^{\mathrm{T}}\mathrm{U}$, $\mathrm{J}^{\mathrm{V}} = \mathrm{V}^{\mathrm{T}}\mathrm{V}$, and variables a, b, c, and d. We adopted the well-known product t-norm ($\top_{\mathrm{prod}}(x, y) \triangleq xy$) and minimum t-norm ($\top_{\min}(x, y) \triangleq \min\{x, y\}$) to define 08BRIp and 08BRIm, respectively.

09EBC is based on the redefinitions of BCP and BCR (Equations 8):

$$\mathrm{EBCP}(\mathrm{U}, \mathrm{V}) \triangleq \frac{1}{n}\sum_{i=1}^{n}\frac{\sum_{j=1}^{n}\min\{\mathrm{J}^{\mathrm{U}}_{i,j}, \mathrm{J}^{\mathrm{V}}_{i,j}\}}{\sum_{j=1}^{n}\mathrm{J}^{\mathrm{U}}_{i,j}} \text{ and} \tag{9a}$$

$$\mathrm{EBCR}(\mathrm{U}, \mathrm{V}) \triangleq \frac{1}{n}\sum_{i=1}^{n}\frac{\sum_{j=1}^{n}\min\{\mathrm{J}^{\mathrm{U}}_{i,j}, \mathrm{J}^{\mathrm{V}}_{i,j}\}}{\sum_{j=1}^{n}\mathrm{J}^{\mathrm{V}}_{i,j}}. \tag{9b}$$

Equations (8) and (9) are equivalent when U and V are EHCs. 09EBC is defined by default as:

$$09\mathrm{EBC}(\mathrm{U}, \mathrm{V}) \triangleq 2 \cdot \frac{\mathrm{EBCP}(\mathrm{U}, \mathrm{V}) \cdot \mathrm{EBCR}(\mathrm{U}, \mathrm{V})}{\mathrm{EBCP}(\mathrm{U}, \mathrm{V}) + \mathrm{EBCR}(\mathrm{U}, \mathrm{V})},$$

for NEHCs U and V.

09CRI and 09CARI are based on a reformulation of contingency matrix N, where the sum operator replaces the multiplication operator (i.e., $\dot{\mathrm{N}}_{r,t} \triangleq \sum_{i=1}^{n}(\mathrm{U}_{r,i} + \mathrm{V}_{t,i})$), and the subsequent pairing variable calculation uses an equivalent formulation (in the EHC domain) to that in Equations (2) (Equations (14), (15), (16), and (21) in (Ceccarelli and Maratea, 2009)). 09CRI and 09CARI are obtained by plugging these new pairing variables into Equations (4) and (5), respectively.

09HI is based on similarity calculations between the columns of U and V. Let $\mathrm{R}^{\mathrm{U}}_{i,j} \triangleq 1 - \|\mathrm{U}_{:,i} - \mathrm{U}_{:,j}\|$ and $\mathrm{R}^{\mathrm{V}}_{i,j} \triangleq 1 - \|\mathrm{V}_{:,i} - \mathrm{V}_{:,j}\|$ for all i, j be the similarities between the columns of U and V, where $\|\cdot\|$ is a norm that yields values in $[0, 1]$.[6] The degree of concordance between the distances from U and V defines the measure: $09\mathrm{HI}(\mathrm{U}, \mathrm{V}) \triangleq 1 - \sum_{i<j}|\mathrm{R}^{\mathrm{U}}_{i,j} - \mathrm{R}^{\mathrm{V}}_{i,j}|/(n(n-1)/2)$.

09RI is based on the co-association matrices J^{U} and J^{V}. The 09RI formulation given in Equation (7) of (Rovetta and Masulli, 2009) is incorrect, and Rovetta, S. kindly provided the correct formulation by personal communication, which we repeat here. Given $\mathrm{J}^{\mathrm{U}} = \mathrm{U}^{\mathrm{T}}\mathrm{U}$ and $\mathrm{J}^{\mathrm{V}} = \mathrm{V}^{\mathrm{T}}\mathrm{V}$, the following variables are computed: $\pi \triangleq \sum_{i<j}\mathrm{J}^{\mathrm{U}}_{i,j}\mathrm{J}^{\mathrm{V}}_{i,j}$, $\sigma_{\mathrm{U}} \triangleq \sum_{i<j}\mathrm{J}^{\mathrm{U}}_{i,j}$, and $\sigma_{\mathrm{V}} \triangleq \sum_{i<j}\mathrm{J}^{\mathrm{V}}_{i,j}$. The 09RI measure is then given by $1 + (2\pi - \sigma_{\mathrm{U}} - \sigma_{\mathrm{V}})/\binom{n}{2}$.

09BRI and 09BARI are based on the pairing variables defined in Equations (7). For example, variable a was defined as $(\sum_{i,j=1}^{n}\dot{\mathrm{J}}^{\mathrm{U}}_{i,j}\dot{\mathrm{J}}^{\mathrm{V}}_{i,j} - n)/2$, where the co-association matrices used are normalized: $\dot{\mathrm{J}}^{\mathrm{U}}_{i,j} \triangleq \sum_{r=1}^{k_{\mathrm{U}}}(\mathrm{U}_{r,i}\mathrm{U}_{r,j})/(\|\mathrm{U}_{:,i}\|_{\mathrm{e}}\|\mathrm{U}_{:,j}\|_{\mathrm{e}})$.[7] Plugging these new variables into Equations (4) and (5) yields 09BRI and 09BARI, respectively.

10QRIp and 10QRIm are derived from 08BRIp and 08BRIm, respectively, by normalizing J^{U} and J^{V} such that all diagonal terms equal 1, and letting U and V be PCs. The

6. We adopted the usual Euclidean norm in the experiments.
7. $\|\cdot\|_{\mathrm{e}}$ is the usual Euclidean norm.

rationale behind this normalization is that a diagonal term $\mathrm{J}^{\mathrm{U}}_{i,i}$ should always provide the maximum, as it somehow represents the degree to which object x_i is in the same cluster as itself.

The 10ARI and 10AARI pairing variables are defined using the original formulation $\mathrm{N} = \mathrm{U}\mathrm{V}^{\mathrm{T}}$ and Equations (2). Equations (4) and (5) are then applied to yield 10ARI and 10AARI, respectively. Anderson et al. (2010) noticed that at least 10ARI does not provide evaluations confined in the interval $[0, 1]$ (as RI does) for general PCs. They thus proposed the use of a normalized contingency matrix $\hat{\mathrm{N}} \triangleq \big(n/\mathrm{N}_{+,+}\big)\mathrm{N}$ to have $\hat{\mathrm{N}}_{+,+} = n$ to alleviate the above issue. We denote the normalized versions of 10ARI and 10AARI by 10ARIn and 10AARIn, respectively.

It has been observed that 10ARIn and 10AARIn do not attain their maxima whenever two equivalent solutions[8] are compared (Anderson et al., 2011). 11ARInm and 11AARInm were then defined to address this issue as:

$$\mathrm{11ARInm}(\mathrm{U}, \mathrm{V}) \triangleq \mathrm{10ARIn}(\mathrm{U}, \mathrm{V}) / \max\{\mathrm{10ARIn}(\mathrm{U}, \mathrm{U}), \mathrm{10ARIn}(\mathrm{V}, \mathrm{V})\} \text{ and}$$
$$\mathrm{11AARInm}(\mathrm{U}, \mathrm{V}) \triangleq \mathrm{10AARIn}(\mathrm{U}, \mathrm{V}) / \max\{\mathrm{10AARIn}(\mathrm{U}, \mathrm{U}), \mathrm{10AARIn}(\mathrm{V}, \mathrm{V})\}.$$

The 10CSI measure was designed to handle non-exclusive and exclusive hard clusterings. Let $\mathrm{J}^{\mathrm{U}} = \mathrm{U}^{\mathrm{T}}\mathrm{U}$ and $\mathrm{J}^{\mathrm{V}} = \mathrm{V}^{\mathrm{T}}\mathrm{V}$ be the co-association matrices, and let $\mathrm{U}_{+,i}$ and $\mathrm{V}_{+,i}$ be the number of clusters to which object x_i belongs, according to the respective solutions. The agreement and disagreement between U and V according to the relative placement of objects x_i and x_j are defined by 10CSI as:

$$a^g_{i,j} \triangleq \min\{\mathrm{J}^{\mathrm{U}}_{i,j}, \mathrm{J}^{\mathrm{V}}_{i,j}\} + \min\{\mathrm{U}_{+,i}, \mathrm{V}_{+,i}\} + \min\{\mathrm{U}_{+,j}, \mathrm{V}_{+,j}\} - 2 \text{ and}$$
$$d^g_{i,j} \triangleq |\mathrm{J}^{\mathrm{U}}_{i,j} - \mathrm{J}^{\mathrm{V}}_{i,j}| + |\mathrm{U}_{+,i} - \mathrm{V}_{+,i}| + |\mathrm{U}_{+,j} - \mathrm{V}_{+,j}|$$

10CSI is given by $\sum_{i<j} a^g_{i,j} / \sum_{i<j}(a^g_{i,j} + d^g_{i,j})$, which reduces to JI in the EHC domain.

The 10CF and 10CFn measures largely differ from the others because they are not pair-based nor based on information theory. 10CF and 10CFn are somehow related to the edit distance commonly used to define the compatibility degree between two strings of text (Levenshtein, 1966). Campello (2010) defined the fuzzy transfer distance $\mathrm{F}_{\mathrm{TD}}(\mathrm{U}, \mathrm{V})$ between two PCs U and V as the minimum amount of membership degrees that must be given to and/or removed from the objects of U (V) to make this clustering equivalent to V (U). We define here 10CF as $\mathrm{10CF}(\mathrm{U}, \mathrm{V}) \triangleq 1 - \mathrm{F}_{\mathrm{TD}}(\mathrm{U}, \mathrm{V})$ such that it yields values in the interval $(-\infty, 1]$ and attains 1 iff U and V are equivalent clusterings (Campello, 2010). 10CFn is 1 minus the normalized version of F_{TD}: $\mathrm{10CFn}(\mathrm{U}, \mathrm{V}) \triangleq 1 - \mathrm{F}_{\mathrm{TD}}(\mathrm{U}, \mathrm{V})/(n \max\{k_{\mathrm{U}}, k_{\mathrm{V}}\})$. $\mathrm{10CFn}(\mathrm{U}, \mathrm{V})$ lies in the interval $[0, 1]$ (Campello, 2010).

Let U and V be two NEHCs with k_{U} and k_{V} clusters, respectively. The 11MD and 11D2 measures are defined as:

$$\mathrm{11MD}(\mathrm{U}, \mathrm{V}) \triangleq 1 - \frac{1}{n}\sum_{i=1}^{n} \frac{\sum_{j=1}^{n} |\mathrm{J}^{\mathrm{U}}_{i,j} - \mathrm{J}^{\mathrm{V}}_{i,j}|}{\sum_{j=1}^{n} \max\{\mathrm{J}^{\mathrm{U}}_{i,j}, \mathrm{J}^{\mathrm{V}}_{i,j}\}} \text{ and}$$
$$\mathrm{11D2}(\mathrm{U}, \mathrm{V}) \triangleq 1 - \frac{1}{n^2}\sum_{i,j=1}^{n} |\mathrm{J}^{\mathrm{U}}_{i,j} - \mathrm{J}^{\mathrm{V}}_{i,j}|,$$

8. Clusterings U and V are equivalent iff (i) they have the same number of clusters and (ii) V can always be transformed into U by row permutations.

where $\sum_{i,j=1}^{n} |\mathrm{J}^{\mathrm{U}}_{i,j} - \mathrm{J}^{\mathrm{V}}_{i,j}|$ is the Hamming distance when U and V are EHCs.

Let A^{U} be the adjacency matrix of U defined as:

$$\mathrm{A}^{\mathrm{U}}_{i,j} \triangleq \begin{cases} 1 & \exists r : \mathrm{U}_{r,i}\mathrm{U}_{r,j} = 1 \\ 0 & \text{otherwise} \end{cases}.$$

The normalized disconnectivity of U is given by default as (Wang, 2012):

$$\mathrm{NDisc}(\mathrm{U}) \triangleq 2(1 - \frac{1}{n^2}\sum_{i,j=1}^{n} \mathrm{A}^{\mathrm{U}}_{i,j}).$$

Let R be a (possibly degenerate) clustering resulting from the intersection between the clusters of NEHCs U and V:

$$\mathrm{R}_{(r+(t-1)*k_{\mathrm{U}}),i} \triangleq \mathrm{U}_{r,i}\mathrm{V}_{t,i}.$$

The 12DB measure is defined by default as:

$$12\mathrm{DB}(\mathrm{U},\mathrm{V}) \triangleq 2 \cdot \mathrm{NDisc}(\mathrm{R}) - \mathrm{NDisc}(\mathrm{U}) - \mathrm{NDisc}(\mathrm{V}).$$

3.2 Discussion

Some authors extended pair-based measures by simply letting U and V be representations of other clustering types (i.e., others than EHC types) in the definition of contingency matrix N (e.g., Ceccarelli and Maratea, 2009; Anderson et al., 2010) or co-association matrices J^{U} and J^{V} (e.g., Borgelt and Kruse, 2006; Borgelt, 2007; Quere and Frelicot, 2011), and computing a, b, c, and d based on Equations (2) or Equations (7). However, the pairing variable equations were deduced by assuming that U and V are EHCs. Without a more principled explanation, we believe there is no reason to expect that using the same definitions would grant meaningful values to a, b, c, and d in more general circumstances. Consider the following identical EHCs:

$$\mathrm{U} \triangleq \mathrm{V} \triangleq \begin{pmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{pmatrix}. \tag{10}$$

We have $a = 0$, $b = 0$, $c = 0$, and $d = 1$, according to the definitions given by Equations (2) and Equations (7). There is only one pair of objects, and the objects are not clustered together in both solutions. Now let

$$\dot{\mathrm{V}} \triangleq \begin{pmatrix} 0.9 & 0.0 \\ 0.1 & 1.0 \end{pmatrix} \tag{11}$$

be an FC very similar to V. Comparing U and $\dot{\mathrm{V}}$, we now have $a = -0.09$, $b = 0.1$, $c = 0.09$, and $d = 0.9$, according to Equations (2), and $a = 0$, $b = 0$, $c = -0.1$, and $d = 0.9$, according to Equations (7). It is hard to assign a meaningful interpretation when a pairing variable yields a negative value. Moreover, the obtained values are no longer equivalent to each other. This result shows that the application of Equations (2) and (7) in more general settings must indeed be accompanied by a good justification.

None of the measures 03VI, 03MI, 05MI, 07CRI, 07CARI, 08BRIp, 08BRIm, 09CRI, 09CARI, 09RI, 09BRI, 09BARI, 10QRIp, 10QRIm, 10ARI, 10AARI, 10ARIn, 10AARIn, 10CF, 11ARInm, and 12DB attain their maxima 1 whenever two equivalent solutions are compared, as Section 7.1 shows. This makes interpreting the evaluation provided by these measures difficult. Moreover, there is no reason to expect that ARI generalizations (i.e., 07CARI, 09CARI, 09BARI, 10AARI, 10AARIn, and 11AARInm) are corrected for randomness in others than in EHC scenarios simply because the original ARI has this property for EHCs (this belief is confirmed in the experiments in Section 7.2). The formulations upon which these generalized measures are based were deduced by assuming that the compared solutions are EHCs.

4. Frand Index

Given two FCs U (with k_{U} clusters) and V (with k_{V} clusters) of n objects, 13FRI recasts each into two n-by-n matrices to retain only the essential information and to facilitate the comparison. Let $\mathrm{I}_{k_{\mathrm{U}}}$ be the k_{U}-by-k_{U} identity matrix and $\mathbb{1}_{k_{\mathrm{U}}}$ be the k_{U}-by-k_{U} matrix with 1 in each entry. Define the matrices

$$\mathrm{J}^{\mathrm{U}} \triangleq \mathrm{U}^{\mathrm{T}}\mathrm{U} \quad \text{and} \tag{12a}$$

$$\mathrm{S}^{\mathrm{U}} \triangleq \mathrm{U}^{\mathrm{T}}(\mathbb{1}_{k_{\mathrm{U}}} - \mathrm{I}_{k_{\mathrm{U}}})\mathrm{U}. \tag{12b}$$

Matrices J^{U} and S^{U} provide all pairwise information between objects for 13FRI with respect to U. Let J^{V} and S^{V} be the corresponding matrices for V. 13FRI compares J^{U} and S^{U} with J^{V} and S^{V} to measure how much U and V agree with the membership assignment of each object pair. Let us elaborate these matrices.

$\mathrm{J}^{\mathrm{U}}_{i,j}$ and $\mathrm{S}^{\mathrm{U}}_{i,j}$ can be interpreted in several ways. For EHCs, $\mathrm{J}^{\mathrm{U}}_{i,j} = 1$ (implying $\mathrm{S}^{\mathrm{U}}_{i,j} = 0$) means that objects x_i and x_j belong to the same cluster in solution U, and $\mathrm{J}^{\mathrm{U}}_{i,j} = 0$ (implying $\mathrm{S}^{\mathrm{U}}_{i,j} = 1$) means that they belong to different clusters in U. In the EHC domain, J^{U} is the same matrix as that defined in Equation (6), and $\mathrm{S}^{\mathrm{U}}_{i,j} = 1 - \mathrm{J}^{\mathrm{U}}_{i,j}$.

Another interpretation can be provided for J^{U} and S^{U} in the FC domain. If one considers that an FC U produces probabilities of objects pertaining to clusters (e.g., as in EM solutions), i.e., $\mathrm{U}_{r,i}$ is the probability of object x_i belonging to the rth cluster, $\mathrm{J}^{\mathrm{U}}_{i,j}$ gives the probability of objects x_i and x_j belonging to the same cluster according to U, and $\mathrm{S}^{\mathrm{U}}_{i,j} = 1 - \mathrm{J}^{\mathrm{U}}_{i,j}$ gives the probability that they belong to different clusters according to U, assuming independence.

We also allow J^{U} and S^{U} to be defined for PCs in general (Section 5). Let us thus consider two other interpretations for J^{U} and S^{U} in the PC domain. Letting U be an NEHC, $\mathrm{J}^{\mathrm{U}}_{i,j}$ is the number of times x_i and x_j belong to the same cluster in U, and $\mathrm{S}^{\mathrm{U}}_{i,j}$ is the number of times x_i and x_j belong to different clusters in U. If U is a more general PC, we can say that $\mathrm{J}^{\mathrm{U}}_{i,j}$ is the possibility of x_i and x_j belonging to the same cluster in U, and $\mathrm{S}^{\mathrm{U}}_{i,j}$ is the possibility of x_i and x_j belonging to different clusters in U.

Despite the above multitude of interpretations, we understand that $\mathrm{J}^{\mathrm{U}}_{i,j}$ represents a degree of truthiness for the sentence "x_i and x_j belong to the same cluster", whereas $\mathrm{S}^{\mathrm{U}}_{i,j}$ yields a degree of falseness to the same sentence, according to the solution U. This reasoning

led us to redefine the pairing variables a, b, c, and d as follows:

$$\dot{a} \triangleq \sum_{i<j} \min\{\mathrm{J}^{\mathrm{U}}_{i,j}, \mathrm{J}^{\mathrm{V}}_{i,j}\}, \tag{13a}$$

$$\dot{b} \triangleq \sum_{i<j} \min\{\mathrm{J}^{\mathrm{U}}_{i,j} - \min\{\mathrm{J}^{\mathrm{U}}_{i,j}, \mathrm{J}^{\mathrm{V}}_{i,j}\}, \mathrm{S}^{\mathrm{V}}_{i,j} - \min\{\mathrm{S}^{\mathrm{U}}_{i,j}, \mathrm{S}^{\mathrm{V}}_{i,j}\}\}, \tag{13b}$$

$$\dot{c} \triangleq \sum_{i<j} \min\{\mathrm{J}^{\mathrm{V}}_{i,j} - \min\{\mathrm{J}^{\mathrm{U}}_{i,j}, \mathrm{J}^{\mathrm{V}}_{i,j}\}, \mathrm{S}^{\mathrm{U}}_{i,j} - \min\{\mathrm{S}^{\mathrm{U}}_{i,j}, \mathrm{S}^{\mathrm{V}}_{i,j}\}\}, \text{ and} \tag{13c}$$

$$\dot{d} \triangleq \sum_{i<j} \min\{\mathrm{S}^{\mathrm{U}}_{i,j}, \mathrm{S}^{\mathrm{V}}_{i,j}\}. \tag{13d}$$

Variables $\dot{a}$ and $\dot{d}$ measure the agreement between U and V with respect to the truthiness and falseness of sentence "x_i and x_j belong to the same cluster" for each pair of objects x_i and x_j; $\dot{b}$ and $\dot{c}$ measure the disagreement. For EHCs U and V, $\min\{\mathrm{J}^{\mathrm{U}}_{i,j}, \mathrm{J}^{\mathrm{V}}_{i,j}\} = 1$ means that x_i and x_j are clustered together in both clusterings. Conversely, $\min\{\mathrm{S}^{\mathrm{U}}_{i,j}, \mathrm{S}^{\mathrm{V}}_{i,j}\} = 1$ means that x_i and x_j belong to different clusters in both clusterings. In both cases, $\dot{a} + \dot{d}$ increases by 1. $\mathrm{J}^{\mathrm{U}}_{i,j} \neq \mathrm{J}^{\mathrm{V}}_{i,j}$ means that there is a disagreement between U and V regarding the pairing of x_i and x_j; it implies that $\min\{\mathrm{J}^{\mathrm{U}}_{i,j}, \mathrm{J}^{\mathrm{V}}_{i,j}\} = \min\{\mathrm{S}^{\mathrm{U}}_{i,j}, \mathrm{S}^{\mathrm{V}}_{i,j}\} = 0$ and increments $\dot{b} + \dot{c}$ by 1. This behavior recalls the descriptive definition of a, b, c, and d given in Section 3. Comparing the definitions in Equations (7) with those in Equations (13), $a = \dot{a}$, $b = \dot{b}$, $c = \dot{c}$, and $d = \dot{d}$ when comparing EHCs. Consequently, our similarity measure

$$\mathrm{13FRI}(\mathrm{U}, \mathrm{V}) \triangleq \frac{\dot{a} + \dot{d}}{\dot{a} + \dot{b} + \dot{c} + \dot{d}} \tag{14}$$

reduces to RI when U and V are EHCs.

Now, consider the more general context where U and V are FCs. We defined $\dot{a} + \dot{d}$ ($\dot{b} + \dot{c}$) to measure to what extent U and V agree (disagree) with each other regarding the object pairings. For example, the min operator in $\min\{\mathrm{S}^{\mathrm{U}}_{i,j}, \mathrm{S}^{\mathrm{V}}_{i,j}\}$ appears to provide a reasonable notion to what extent the solutions agree that x_i and x_j should not be clustered together. When the elements of J^{U} and J^{V} (or S^{U} and S^{V}) simultaneously show high or low values, there is a strong compatibility between U and V. This is reflected by how 13FRI was defined.

One may ask why $\dot{b}$ (and similarly for $\dot{c}$) was not defined as $\dot{b} \triangleq \sum_{i<j} \min\{\mathrm{J}^{\mathrm{U}}_{i,j}, \mathrm{S}^{\mathrm{V}}_{i,j}\}$. The reason is that the amount $\min\{\mathrm{J}^{\mathrm{U}}_{i,j}, \mathrm{J}^{\mathrm{V}}_{i,j}\}$ has already been used from $\mathrm{J}^{\mathrm{U}}_{i,j}$ and $\mathrm{J}^{\mathrm{V}}_{i,j}$ to establish the agreement between $\mathrm{J}^{\mathrm{U}}_{i,j}$ and $\mathrm{J}^{\mathrm{V}}_{i,j}$ in $\dot{a}$. Suppose that $\mathrm{J}^{\mathrm{U}}_{i,j} = \mathrm{S}^{\mathrm{U}}_{i,j} = \mathrm{J}^{\mathrm{V}}_{i,j} = \mathrm{S}^{\mathrm{V}}_{i,j} = x$. Let $\dot{a}_{i,j} \triangleq \min\{\mathrm{J}^{\mathrm{U}}_{i,j}, \mathrm{J}^{\mathrm{V}}_{i,j}\}$, and analogously define $\dot{b}_{i,j}$, $\dot{c}_{i,j}$, and $\dot{d}_{i,j}$. Without the subtractions in Equations (13b) and (13c), each variable $\dot{a}$, $\dot{b}$, $\dot{c}$, and $\dot{d}$ would be increased by x (i.e., $\dot{a}_{i,j} = \dot{b}_{i,j} = \dot{c}_{i,j} = \dot{d}_{i,j} = x$), meaning that U and V would have only 50% agreement regarding the placement of x_i and x_j, instead of 100%. This does not happen with the original formulation because all the information regarding the placement of x_i and x_j has been used in the definition of $\dot{a}_{i,j}$ and $\dot{d}_{i,j}$, and then nothing is left to the definition of $\dot{b}_{i,j}$ and $\dot{c}_{i,j}$. Figure 2 represents the values $\mathrm{J}^{\mathrm{U}}_{i,j} + \mathrm{S}^{\mathrm{U}}_{i,j} = 2x$ and $\mathrm{J}^{\mathrm{V}}_{i,j} + \mathrm{S}^{\mathrm{V}}_{i,j} = 2x$ by box heights. Parallel line orientations define the two types of filled areas regarding the information used

from the co-association matrices to determine $\dot{a}_{i,j}$ and $\dot{d}_{i,j}$. There is no space in the boxes (i.e., unused information) to fill regarding variables $\dot{b}_{i,j}$ and $\dot{c}_{i,j}$.

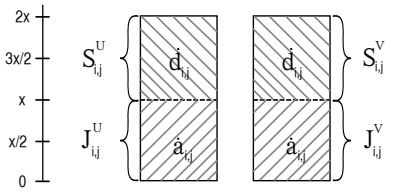


Figure 2: Graphical representation of a 13FRI evaluation where $\dot{b}_{i,j} = \dot{c}_{i,j} = 0$.

The 13FRI measure yields values in the continuous interval $[0, 1]$. It attains the maximum 1 whenever equivalent solutions are compared[9] and attains the minimum 0 only when U and V are EHCs and one of them has one cluster and the other has n clusters (Proposition 1 in Appendix). However, this last scenario is extreme and has little practical value (Vinh et al., 2009, 2010), making low 13FRI evaluations nearly impossible in practice. It is desirable that the entire interval $[0, 1]$ be useful, for better intuitiveness. This can be achieved by a similarity measure that takes values close to a constant α (α can always be turned into zero by a non-linear transformation: subtracting α from the evaluation and multiplying the result by a β that makes the maximum equals 1) when comparing random solutions (constant baseline). When a constant baseline exists and the user knows its value beforehand, one can compare the obtained evaluation to the baseline value and be more confident in his conclusions. The next section shows how 13FRI can be adjusted to assume values close to zero for randomly generated solutions.

4.1 Adjustment for Randomness

Suppose a measure assigns x to the similarity between two FCs U and V. How can we determine if x is not just a value from the random fluctuation inherent to the measure? A popular approach addresses this issue by subtracting the measure expectation from the measure and normalizing the result to 1 as a maximum (Hubert and Arabie, 1985; Albatineh et al., 2006; Vinh et al., 2009, 2010):

$$\mathrm{ASM}(\mathrm{U},\mathrm{V}) \triangleq \frac{\mathrm{SM}(\mathrm{U},\mathrm{V}) - \mathrm{E}[\mathrm{SM}]_{\mathrm{U},\mathrm{V}}}{\max\{\mathrm{SM}\} - \mathrm{E}[\mathrm{SM}]_{\mathrm{U},\mathrm{V}}}, \tag{15}$$

where SM is any similarity measure, $\mathrm{E}[\mathrm{SM}]_{\mathrm{U},\mathrm{V}}$ is its expectation given U and V, $\max\{\mathrm{SM}\}$ is the maximum of SM, and ASM is its adjusted version. ASM assumes values in the range $(-\infty, 1]$, and a positive value indicates that the similarity between U and V is greater than what one would expect from randomly chosen solutions. As Section 7.2 indicates for our corrected measures, this adjustment for chance can also make the measure unbiased in the number of clusters (Vinh et al., 2009, 2010).

To correct a measure for randomness, it is necessary to specify a null model according to which solutions are generated (Vinh et al., 2009, 2010). Given two FCs U and V, our

9. Note that $\mathrm{J}^{\mathrm{U}}_{i,j}$ and $\mathrm{S}^{\mathrm{U}}_{i,j}$ are independent of U row permutations. If U and V are equivalent clusterings, we have $\mathrm{J}^{\mathrm{U}}_{i,j} = \mathrm{J}^{\mathrm{V}}_{i,j}$ and $\mathrm{S}^{\mathrm{U}}_{i,j} = \mathrm{S}^{\mathrm{V}}_{i,j}$ $\forall i < j$. It implies that $\dot{b} = \dot{c} = 0$ and $\mathrm{13FRI}(\mathrm{U},\mathrm{V}) = 1$.

null model simultaneously produces two solutions from independent random permutations of the U and V columns. Let $\pi_1, \pi_2, \ldots, \pi_{n!}$ be every possible permutation of the numbers in $\mathbb{N}_{1,n}$, and define the function $\Gamma_{\pi_l}(\mathrm{U}) \triangleq [\mathrm{U}_{:,\pi_l(1)} \quad \mathrm{U}_{:,\pi_l(2)} \quad \ldots \quad \mathrm{U}_{:,\pi_l(n)}]$ that applies permutation π_l to matrix U.[10] A particular permutation π_l of U is chosen with probability $\mathrm{P}(\pi_l) \triangleq 1/n!$, and the permutations of U and V are considered independent events. We thus define $\mathrm{P}(\pi_l, \pi_q) \triangleq 1/(n!n!)$. The expectation of 13FRI according to our null model given U and V is

$$\mathrm{E}[13\mathrm{FRI}]_{\mathrm{U,V}} = \frac{1}{n!n!} \sum_{l,q=1}^{n!} 13\mathrm{FRI}(\Gamma_{\pi_l}(\mathrm{U}), \Gamma_{\pi_q}(\mathrm{V})). \tag{16}$$

Let $\dot{a}(\mathrm{J}^{\mathrm{U}}, \mathrm{J}^{\mathrm{V}}) \triangleq \sum_{i<j} \min\{\mathrm{J}^{\mathrm{U}}_{i,j}, \mathrm{J}^{\mathrm{V}}_{i,j}\}$ and $\dot{d}(\mathrm{S}^{\mathrm{U}}, \mathrm{S}^{\mathrm{V}}) \triangleq \sum_{i<j} \min\{\mathrm{S}^{\mathrm{U}}_{i,j}, \mathrm{S}^{\mathrm{V}}_{i,j}\}$. Because $\dot{a} + \dot{b} + \dot{c} + \dot{d}$ is a constant for the proposed null model (Corollary 1 in Appendix), we rewrite the expectation

$$\mathrm{E}[13\mathrm{FRI}]_{\mathrm{U,V}} = (\dot{a} + \dot{b} + \dot{c} + \dot{d})^{-1}(\mathrm{E}[\dot{a}]_{\mathrm{U,V}} + \mathrm{E}[\dot{d}]_{\mathrm{U,V}}), \tag{17}$$

where

$$\begin{aligned}
\mathrm{E}[\dot{a}]_{\mathrm{U,V}} &= \frac{1}{n!n!} \sum_{l,q=1}^{n!} \dot{a}(\mathrm{J}^{\Gamma_{\pi_l}(\mathrm{U})}, \mathrm{J}^{\Gamma_{\pi_q}(\mathrm{V})}) \\
&= \frac{1}{n!n!} \sum_{l,q=1}^{n!} \sum_{i_1<j_1} \min\{\mathrm{J}^{\mathrm{U}}_{\pi_l(i_1),\pi_l(j_1)}, \mathrm{J}^{\mathrm{V}}_{\pi_q(i_1),\pi_q(j_1)}\} \\
&= \frac{2(n-2)!}{n!n!} \sum_{q=1}^{n!} \sum_{i_1<j_1} \sum_{i_2<j_2} \min\{\mathrm{J}^{\mathrm{U}}_{i_2,j_2}, \mathrm{J}^{\mathrm{V}}_{\pi_q(i_1),\pi_q(j_1)}\} \\
&= \frac{2(n-2)!2(n-2)!}{n!n!} \sum_{i_1<j_1} \sum_{i_2<j_2} \sum_{i_3<j_3} \min\{\mathrm{J}^{\mathrm{U}}_{i_2,j_2}, \mathrm{J}^{\mathrm{V}}_{i_3,j_3}\} \\
&= \frac{4}{n^2(n-1)^2} \sum_{i_1<j_1} \sum_{i_2<j_2} \sum_{i_3<j_3} \min\{\mathrm{J}^{\mathrm{U}}_{i_2,j_2}, \mathrm{J}^{\mathrm{V}}_{i_3,j_3}\} \\
&= \frac{2}{n(n-1)} \sum_{i_2<j_2} \sum_{i_3<j_3} \min\{\mathrm{J}^{\mathrm{U}}_{i_2,j_2}, \mathrm{J}^{\mathrm{V}}_{i_3,j_3}\}
\end{aligned} \tag{18}$$

and, analogously,

$$\mathrm{E}[\dot{d}]_{\mathrm{U,V}} = \frac{2}{n(n-1)} \sum_{i_2<j_2} \sum_{i_3<j_3} \min\{\mathrm{S}^{\mathrm{U}}_{i_2,j_2}, \mathrm{S}^{\mathrm{V}}_{i_3,j_3}\}. \tag{19}$$

Following the framework of Equation (15), the adjusted frand index is

$$13\mathrm{AFRI}(\mathrm{U,V}) \triangleq \frac{13\mathrm{FRI}(\mathrm{U,V}) - \mathrm{E}[13\mathrm{FRI}]_{\mathrm{U,V}}}{1 - \mathrm{E}[13\mathrm{FRI}]_{\mathrm{U,V}}}. \tag{20}$$

10. $\mathrm{U}_{:,i}$ is the ith column of U.

13AFRI attains its maximum 1 in the same way as 13FRI (i.e., whenever two equivalent clusterings are compared) and is 0 when the measure equals its expected value, under the null model. 13AFRI can display negative evaluations, which mean that the compared clusterings are more dissimilar than expected if they were independently generated. Its minimum is not fixed anymore and is given by $-\mathrm{E[SM]_{U,V}}/(\max\{\mathrm{SM}\} - \mathrm{E[SM]_{U,V}})$.

Given two EHCs U and V, we have $\mathrm{13AFRI(U,V)} = \mathrm{ARI(U,V)}$ (Proposition 3 in Appendix). In other words, 13AFRI reduces to ARI in the EHC domain. This indicates the appropriateness of the null model for 13AFRI, which can also be further extended to PCs (as Section 5 shows).

4.2 Discussion

13FRI could also be applied to PCs. In this case, however, 13FRI would not provide reasonable evaluations in some scenarios where per-object membership totals (i.e., column-wise sums of the clustering matrix) varies among solutions. Let U be an FC and recall that an FC is also a PC. The result of multiplying U by a scalar $x \in (0,1)$ is also a PC matrix, where the per-object membership total of each object is decreased. Notice that we have $\mathrm{13FRI(U,U)} = \mathrm{13FRI(U},x\mathrm{U)} = \mathrm{13AFRI(U,U)} = \mathrm{13AFRI(U},x\mathrm{U)} = 1$ for any $x \in (0,1]$. This happens because $\mathrm{J}^{x\mathrm{U}}_{i,j} = \min\{\mathrm{J}^{\mathrm{U}}_{i,j}, \mathrm{J}^{x\mathrm{U}}_{i,j}\}$ and $\mathrm{S}^{x\mathrm{U}}_{i,j} = \min\{\mathrm{S}^{\mathrm{U}}_{i,j}, \mathrm{S}^{x\mathrm{U}}_{i,j}\}$, making variables $\dot{b}$ and $\dot{c}$ (Equations 13b and 13c) equal to zero.

Let us analyze another problematic scenario by considering the following matrices:

$$\mathrm{U} \triangleq \begin{pmatrix} 0.8 & 0.4 \\ 0.4 & 0.8 \end{pmatrix} \quad \text{and} \quad \mathrm{V} \triangleq \begin{pmatrix} 0.6 & 0.4 \\ 0.4 & 0.6 \end{pmatrix}.$$

Note that U is a PC more general than an FC. We have $\mathrm{J}^{\mathrm{U}}_{1,2} = 0.64$, $\mathrm{S}^{\mathrm{U}}_{1,2} = 0.8$, $\mathrm{J}^{\mathrm{V}}_{1,2} = 0.48$, and $\mathrm{S}^{\mathrm{V}}_{1,2} = 0.52$. The heights of the first and second boxes in Figure 3 correspond to the values $\mathrm{J}^{\mathrm{U}}_{1,2}+\mathrm{S}^{\mathrm{U}}_{1,2} = 1.44$ and $\mathrm{J}^{\mathrm{V}}_{1,2}+\mathrm{S}^{\mathrm{V}}_{1,2} = 1$, respectively. The boxes are divided by horizontal dashed lines, creating two parts that correspond to the $\mathrm{J}^{\mathrm{U}}_{1,2}$ and $\mathrm{S}^{\mathrm{U}}_{1,2}$ ($\mathrm{J}^{\mathrm{V}}_{1,2}$ and $\mathrm{S}^{\mathrm{V}}_{1,2}$) values. The values of $\dot{a} = 0.48$ and $\dot{d} = 0.52$ are illustrated by the filled areas, and the remaining variables $\dot{b}$ and $\dot{c}$ equal zero. There is an empty space of height $\mathrm{J}^{\mathrm{U}}_{1,2}+\mathrm{S}^{\mathrm{U}}_{1,2}-(\mathrm{J}^{\mathrm{V}}_{1,2}+\mathrm{S}^{\mathrm{V}}_{1,2}) = 0.44$ in the first box, which 13FRI ignores. We could increase $\mathrm{J}^{\mathrm{U}}_{1,2}$ and $\mathrm{S}^{\mathrm{U}}_{1,2}$ by any amount that 13FRI would still yield the same score. A reasonable measure for PCs should decrease the score proportionally to the unmatched amount. The next section proposes modifying 13FRI to address this issue.

5. Grand Index

Let $\mathrm{T^U} \triangleq \mathrm{J^U} + \mathrm{S^U}$ and $\mathrm{M} \triangleq \max\{\mathrm{T^U}, \mathrm{T^V}\}$.[11] A new variable

$$\dot{e} \triangleq \max\left\{ \sum_{i<j} \left(\mathrm{M}_{i,j} - \mathrm{T}^{\mathrm{U}}_{i,j}\right), \sum_{i<j} \left(\mathrm{M}_{i,j} - \mathrm{T}^{\mathrm{V}}_{i,j}\right) \right\} \tag{21}$$

11. $\mathrm{M} = \max\{\mathrm{T^U}, \mathrm{T^V}\}$ means that $\mathrm{M}_{i,j} = \max\{\mathrm{T}^{\mathrm{U}}_{i,j}, \mathrm{T}^{\mathrm{V}}_{i,j}\}$ for all i, j.

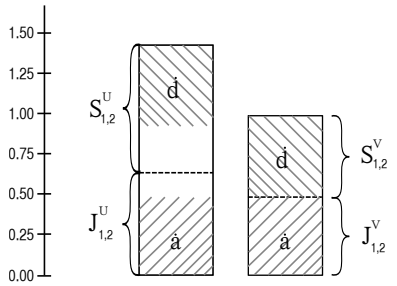


Figure 3: Graphical representation of the problem using 13FRI when the compared clustering matrices have different column-wise sums.

is introduced in 13FRI to give rise to the grand index:

$$\text{13GRI}(\text{U},\text{V}) \triangleq \frac{\dot{a}+\dot{d}}{\dot{a}+\dot{b}+\dot{c}+\dot{d}+\dot{e}}. \tag{22}$$

Given two objects x_i and x_j, $\text{M}_{i,j} - \text{T}^{\text{U}}_{i,j}$ describes how much $\text{T}^{\text{V}}_{i,j}$ exceeds $\text{T}^{\text{U}}_{i,j}$. In Figure 3, $\text{M}_{i,j} - \text{T}^{\text{V}}_{i,j} = 0.44$, which equals the height of the empty space in the first box. Proposition 5 in Appendix allows us to rewrite Equation (22) as

$$\text{13GRI}(\text{U},\text{V}) = \frac{\sum_{i<j} \min\{\text{J}^{\text{U}}_{i,j}, \text{J}^{\text{V}}_{i,j}\} + \sum_{i<j} \min\{\text{S}^{\text{U}}_{i,j}, \text{S}^{\text{V}}_{i,j}\}}{\max\{\sum_{i<j} \text{T}^{\text{U}}_{i,j}, \sum_{i<j} \text{T}^{\text{V}}_{i,j}\}}.$$

If U and V are FCs, $\text{T}^{\text{U}}_{i,j} = \text{T}^{\text{V}}_{i,j} = 1$, $\dot{a}+\dot{b}+\dot{c}+\dot{d}+\dot{e} = \max\{\sum_{i<j} \text{T}^{\text{U}}_{i,j}, \sum_{i<j} \text{T}^{\text{V}}_{i,j}\} = n(n-1)/2$, and 13GRI reduces to 13FRI. As in 13FRI, 13GRI attains its maximum 1 whenever the compared PCs U and V are equivalent solutions.[12]

Adopting the same null model proposed in Section 4.1, and realizing that $\dot{a}+\dot{b}+\dot{c}+\dot{d}+\dot{e}$ is constant for this model (Corollary 2 in Appendix), we have $\text{E}[\text{13GRI}]_{\text{U,V}} = (\dot{a}+\dot{b}+\dot{c}+\dot{d}+\dot{e})^{-1}(\text{E}[\dot{a}]_{\text{U,V}} + \text{E}[\dot{d}]_{\text{U,V}})$. The adjusted 13GRI is then given by

$$\text{13AGRI}(\text{U},\text{V}) \triangleq \frac{\text{13GRI}(\text{U},\text{V}) - \text{E}[\text{13GRI}]_{\text{U,V}}}{1 - \text{E}[\text{13GRI}]_{\text{U,V}}}. \tag{23}$$

Similarly to 13AFRI, 13AGRI attains its maximum 1 in the same way as 13GRI and is 0 when the measure equals its expected value. Section 7.2 shows that 13AGRI can indeed exhibit a constant baseline close to zero for randomly generated EHC, FC, NEHC, and PC solutions, even when the null model is clearly violated.

6. Computational Complexity and Implementation

Let I_{k_U} be the k_U-by-k_U identity matrix and $\mathbb{1}_{k_\text{U}}$ the k_U-by-k_U matrix with 1 in each entry. There are $O(n^2 k_\text{U})$ computational steps to calculate $\text{J}^{\text{U}} = \text{U}^{\text{T}}\text{U}$ and $\text{S}^{\text{U}} = \text{U}^{\text{T}}(\mathbb{1}_{k_\text{U}} - \text{I}_{k_\text{U}})\text{U}$,

12. As in the 13FRI case, we have $\text{J}^{\text{U}}_{i,j} = \text{J}^{\text{V}}_{i,j}$ and $\text{S}^{\text{U}}_{i,j} = \text{S}^{\text{V}}_{i,j}$ $\forall i < j$ whenever U and V are equivalent clusterings. Thus, $\text{T}^{\text{U}}_{i,j} = \text{T}^{\text{V}}_{i,j}$ $\forall i < j$, making $\dot{a}$ and $\dot{d}$ the only possible non-null terms.

and $O(n^2)$ steps to calculate $\mathrm{M} = \max\{\mathrm{T}^{\mathrm{U}}, \mathrm{T}^{\mathrm{V}}\}$. Variables $\dot{a}$, $\dot{b}$, $\dot{c}$, $\dot{d}$, and $\dot{e}$ require $O(n^2)$ steps because of the pairwise summations $\sum_{i<j}$ in their formulas (Equations 13 and 21). 13FRI and 13GRI thus require $O(n^2(k_{\mathrm{U}} + k_{\mathrm{V}}))$ operations. Calculation of EHC pair-based measures generally requires $O(nk_{\mathrm{U}}k_{\mathrm{V}})$ steps due to the contingency matrix $\mathrm{N} = \mathrm{U}\mathrm{V}^{\mathrm{T}}$ computation. The possibly higher 13FRI and 13GRI complexity is the price one may have to pay for a more general measure.

Equations (18) and (19) might suggest that 13AFRI (and 13AGRI) requires $O(n^4)$ computational steps, making its computation infeasible for most practical scenarios. Fortunately, the min operator allows us to reduce the computational complexity of Equations (18) and (19) to $O(n^2 \log n)$ steps. To examine how that can be accomplished, suppose that $\mathrm{J}^{\mathrm{U}}_{1,2} \leq \mathrm{J}^{\mathrm{V}}_{i,j}$ for all $i < j$ $(i, j \in \mathbb{N}_{1,n})$ as a special case and as a didactic example. We have $\sum_{i<j} \min\{\mathrm{J}^{\mathrm{U}}_{1,2}, \mathrm{J}^{\mathrm{V}}_{i,j}\} = \mathrm{J}^{\mathrm{U}}_{1,2} n(n-1)/2$ computable in constant time, reducing the total computational cost. Let us consider the general case for calculating $\mathrm{E}[\dot{a}]_{\mathrm{U,V}}$ (Equation 18). Define

$$\mathbf{1}^{i_2,j_2}_{i_1,j_1} \triangleq \begin{cases} 1 & \text{if } \mathrm{J}^{\mathrm{U}}_{i_1,j_1} \leq \mathrm{J}^{\mathrm{V}}_{i_2,j_2} \\ 0 & \text{otherwise} \end{cases}.$$

Equation (18) can be rewritten as

$$\begin{aligned} \frac{n(n-1)}{2}\mathrm{E}[\dot{a}]_{\mathrm{U,V}} &= \sum_{i_1<j_1}\sum_{i_2<j_2} \min\{\mathrm{J}^{\mathrm{U}}_{i_1,j_1}, \mathrm{J}^{\mathrm{V}}_{i_2,j_2}\}\mathbf{1}^{i_2,j_2}_{i_1,j_1} + \sum_{i_2<j_2}\sum_{i_1<j_1} \min\{\mathrm{J}^{\mathrm{U}}_{i_1,j_1}, \mathrm{J}^{\mathrm{V}}_{i_2,j_2}\}(1 - \mathbf{1}^{i_2,j_2}_{i_1,j_1}) \\ &= \sum_{i_1<j_1} \mathrm{J}^{\mathrm{U}}_{i_1,j_1} \sum_{i_2<j_2} \mathbf{1}^{i_2,j_2}_{i_1,j_1} + \sum_{i_2<j_2} \mathrm{J}^{\mathrm{V}}_{i_2,j_2} \sum_{i_1<j_1} (1 - \mathbf{1}^{i_2,j_2}_{i_1,j_1}). \end{aligned} \tag{24}$$

The calculation of $\mathrm{E}[\dot{d}]_{\mathrm{U,V}}$ (Equation 19) is analogous; the only difference lies in using S^{U} and S^{V} instead of J^{U} and J^{V}.

The above strategy can be applied efficiently by first rearranging the upper triangular parts of J^{U} and J^{V} into vectors x and y, respectively, and sorting the resulting vectors.[13] Algorithm 1 shows an implementation of the above strategy, where the first and second terms of the right-hand side of Equation (24) are calculated by the loops in Steps 7 and 15, respectively.

The most demanding step of Algorithm 1 in terms of computational time is Step 4, which sorts two vectors of size $n(n-1)/2$ in $O(n^2 \log n)$ steps using, for example, the heap sort algorithm. 13AGRI and 13AFRI thus require $O(n^2(k_{\mathrm{U}} + k_{\mathrm{V}} + \log n))$ computational steps.

7. Experiments

It is a common practice to compare the accuracy of clustering algorithms by measuring how similar their resulting clusterings are to a reference solution. The algorithm that generated clusterings more similar to the reference solution is then regarded as the most accurate.

13. The upper triangular part of $\mathrm{J}^{\mathrm{U}}_{i,j}$ can be rearranged as follows: $\mathrm{x}_{\pi(i,j)} \triangleq \mathrm{J}^{\mathrm{U}}_{i,j}$ $(\forall i < j)$, where $\pi(i,j) \triangleq j - i + \sum_{t=1}^{i-1}(n-t) = j - i(1+i)/2 + n(i-1)$.

Algorithm 1 Compute $\mathrm{E}[\dot{a}]_{\mathrm{U,V}}$

Represent the upper triangular part of J^{U} into vector x
Represent the upper triangular part of J^{V} into vector y
$m \leftarrow n(n-1)/2$ {size of vectors x and y}
Sort x and y in increasing order
$\mathrm{E}[\dot{a}]_{\mathrm{U,V}} \leftarrow 0$
$i, j \leftarrow m, m$
while $i > 0$ **do**
  while $j > 0$ **and** $\mathrm{x}_i \leq \mathrm{y}_j$ **do**
    $j \leftarrow j - 1$
  end while
  $\mathrm{E}[\dot{a}]_{\mathrm{U,V}} \leftarrow \mathrm{E}[\dot{a}]_{\mathrm{U,V}} + (m - j) * \mathrm{x}_i$
  $i \leftarrow i - 1$
end while
$i, j \leftarrow m, m$
while $j > 0$ **do**
  while $i > 0$ **and** $\mathrm{x}_i > \mathrm{y}_j$ **do**
    $i \leftarrow i - 1$
  end while
  $\mathrm{E}[\dot{a}]_{\mathrm{U,V}} \leftarrow \mathrm{E}[\dot{a}]_{\mathrm{U,V}} + (m - i) * \mathrm{y}_j$
  $j \leftarrow j - 1$
end while
$\mathrm{E}[\dot{a}]_{\mathrm{U,V}} \leftarrow \mathrm{E}[\dot{a}]_{\mathrm{U,V}}/m$

A measure must somehow adequately evaluate the similarity between the compared solutions. Section 7.1 follows this idea and compares 34 measures by applying them to evaluate solutions with different numbers of clusters produced by different clustering algorithms. This comparison is done by considering the first three properties proposed in Section 2: maximum, discriminant, and contrast. Synthetic data sets were generated according to the cluster types that these algorithms search for (e.g., it is well-known that k-means (MacQueen, 1967) tends to produce spherical-like clusters), and the reference solution for each data set was defined by applying the corresponding clustering algorithm with a well-tuned initial solution. In this scenario is then expected that the dissimilarity between the generated and reference solutions will reflect the difference in the numbers of clusters.

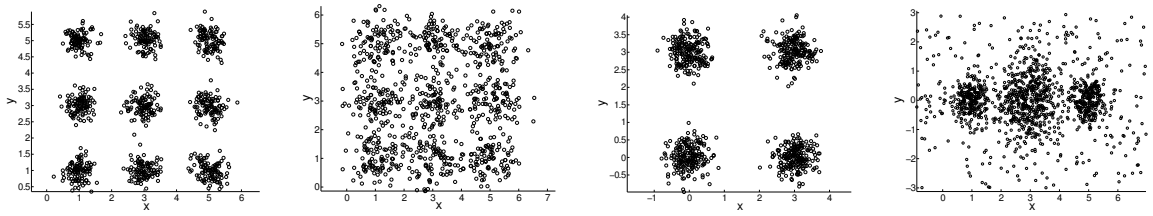
In a different scenario, Section 7.2 compares the measures when evaluating randomly generated solutions, by assessing the measures according to the baseline property proposed in Section 2. A measure should display a uniform evaluation across the range of numbers of clusters because any resemblance between the compared solutions is only due to chance.

Section 7.3 assesses the 13AGRI evaluation validity for FCs in 14 real data sets, and Section 7.4 uses 13AGRI as a stability statistic for estimating the number of clusters in five real data sets.

Because 13GRI (13AGRI) is more general and becomes equivalent to 13FRI (13AFRI) when applied to FCs, we only show the results of 13GRI (13AGRI).

7.1 Measuring the Similarity Between Clusterings

We evaluated the measures in four synthetic data sets (Figures 4), each suitable for one of the following clustering types: EHC, FC, NEHC, and PC. The DEHC data set (Figure 4(a)) has nine well-separated clusters, whereas the DFC data set (Figure 4(b)) has nine overlapping clusters. In both data sets, the clusters were generated using Gaussian distributions with equal variances and no correlation between the attributes. The DNEHC data set (Figure 4(c)) has four clusters, but they reduce to two clusters when projected to a single axis.[14] We generated the DPC data set (Figure 4(d)) to resemble a synthetic one (Zhang and Leung, 2004) with noise added.



(a) Data set for EHC (DEHC).

(b) Data set for FC (DFC).

(c) Data set for NEHC (DNEHC).

(d) Data set for PC (DPC).

Figure 4: Data set for each clustering type.

Different clustering algorithms were employed for each data set, appropriate for the corresponding clustering type as follows: k-means for DEHC, fuzzy c-means (FCM) and expectation maximization for Gaussian mixtures (EMGM) (Dempster et al., 1977) for DFC, SUBCLU (Kailing et al., 2004) for DNEHC, and improved possibilistic c-means 2 (IPCM2) (Zhang and Leung, 2004) for DPC. The FCM and IPCM2 exponent m was set to 2 (which is commonly adopted in the literature), the SUBCLU parameter *minpts* was set to 5, and the Euclidean norm was adopted; this same configuration was used in all the experiments reported in this work. The reference solution for the combination of data set and clustering algorithm (i.e., (DEHC, k-means), (DFC, FCM), (DFC, EMGM), (DNEHC, SUBCLU), and (DPC, IPCM2)) was produced by applying the clustering algorithm with the right number of clusters (or a well-tuned epsilon for SUBCLU), and the result was analyzed to ensure that the solution could be considered ideal in the clustering space sought by the corresponding algorithm. For example, we applied k-means to DEHC with $k = 9$ clusters, using the means of the Gaussian distributions (used to generate the clusters) as the initial centroids. The final solution had virtually the same initial centroids, corroborating the validity of the obtained solution.

It is worth noting that we are not suggesting that the considered clustering algorithms are not suitable for the data sets to which they have not been applied to. For example, FCM can easily find the clustering structure in DEHC, as well as IPCM2 can find the clustering structure in DFC. What is most important is that the data set has a clustering structure suitable for the clustering algorithm being applied.

14. The other data sets could have a similar interpretation as well. However, we only consider subspaces in this specific data set.

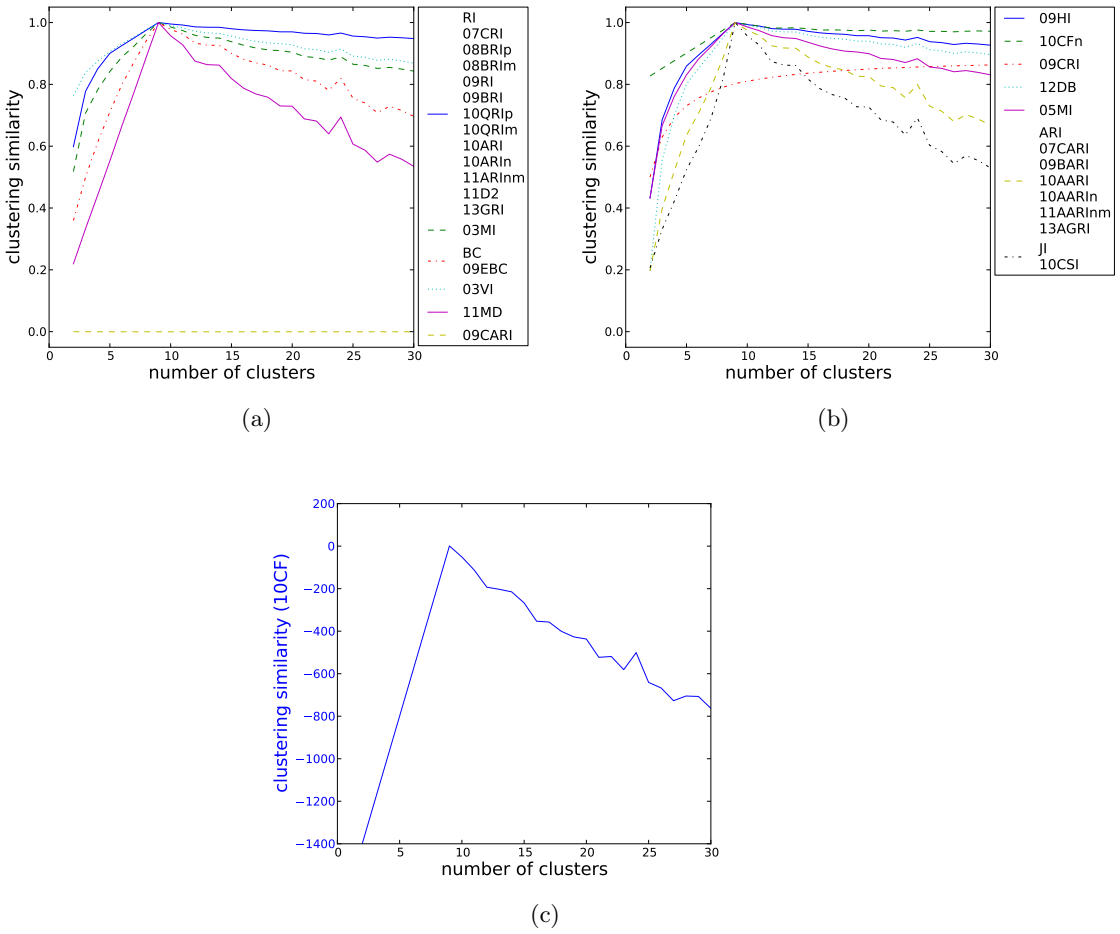


Figure 5: EHC measure evaluations of k-means solutions for the DEHC data set.

The algorithms k-means, FCM, EMGM, and IPCM2 were applied 30 times for each number of clusters $k \in \{2, 3, \ldots, \sqrt{n}\}$ (the literature commonly adopts the upper threshold $\sqrt{n}$ as a rule of thumb (Pal and Bezdek, 1995; Pakhira et al., 2005)), and SUBCLU was applied 30 times for each epsilon in the range $\{0.1, 0.2, \ldots, 5.0\}$. The measures were applied to each solution, and only the highest (which means "the best") values attained in each k or epsilon for a given measure were retained to generate the plots in Figures 5, 6, 7, 8, and 9. We opted to plot the highest values instead of averages because we are interested in the solutions that are as close as possible to the reference one, for a given number of clusters (or epsilon), and to make the results as independent as possible to the stochastic nature of the algorithms. Measures showing the same values were joined and represented by a single curve, and multiple figures for the same experiments were plotted for visualization purposes.

Figure 5 shows that most generalized measures displayed the same results as RI or ARI, when evaluating EHCs. This is expected because most of these measures were defined

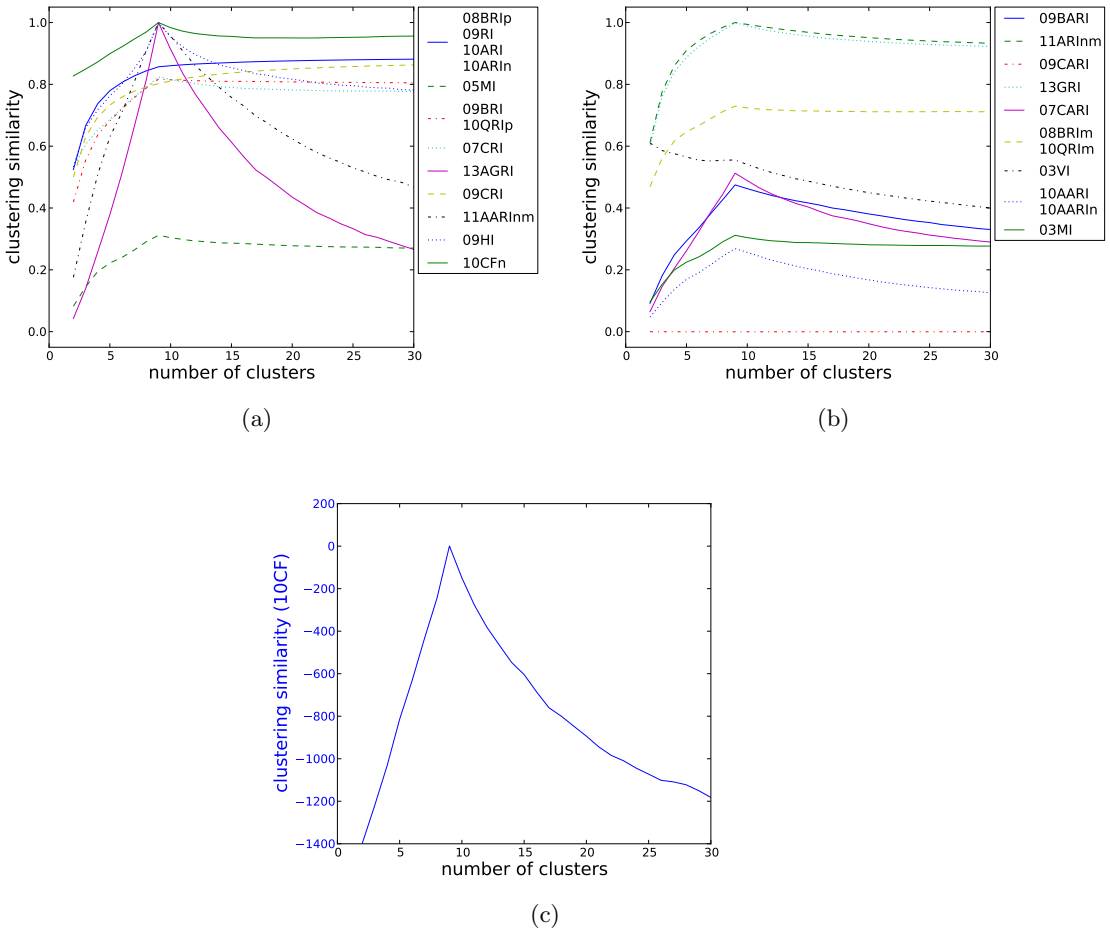


Figure 6: FC measure evaluations of FCM solutions for the DFC data set.

by extending the variables behind the RI or ARI formulations. For example, the 07CRI measure is a fuzzy version of RI in which the pairing variables a, b, c, and d were defined using fuzzy sets. When applied to EHCs, 07CRI reduces to RI (Campello, 2007). RI, 09HI, 10CFn, 12DB, and the measures that showed the same results as RI were weakly affected by a positive difference between the obtained and the true numbers of clusters. RI is equal to 1 and 0.94 for the solutions with 9 and 30 clusters, respectively, which represents less than 10% of its total range $[0, 1]$. This weak responsiveness to the number of clusters makes it difficult to decide whether the solution at hand is really good or not (weak contrast property). 09CRI exhibited an increasing evaluation across the numbers of clusters, and 09CARI produced scores close to zero only. In fact, 09CARI resulted in evaluations close to zero for each scenario in this section. Conversely, JI, ARI, BC, 09EBC, 10CSI, 11MD, and the measures that showed the same results as ARI (including 13AGRI proposed here) exhibited a steady decrease for high numbers of clusters. We believe that this more prominent responsiveness

to differences in the clusterings is more intuitively appealing. 10CF (Figure 5(c)) attained the maximum 1 for the right number of clusters.

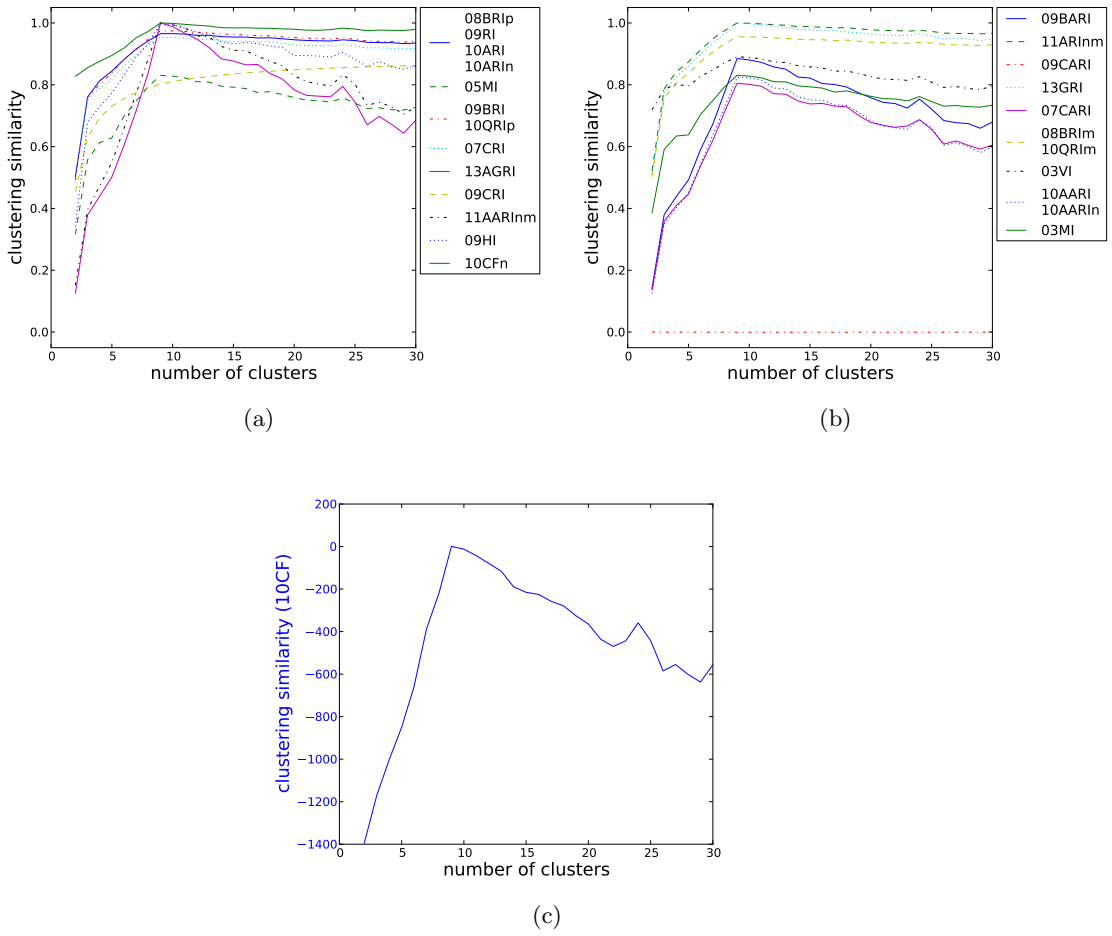


Figure 7: FC measure evaluations of EMGM solutions for the DFC data set.

Figure 6 shows FC measure evaluations of FCM solutions for the DFC data set. Only 13AGRI and 11AARInm provided both the maximum value 1 for the true number of clusters and showed steady decreasing evaluations over the positive increase in the difference between the obtained and true numbers of clusters. 09HI was 1 for the true number of clusters, but it showed an asymptotic-like curve for high numbers of clusters. 03VI, 08BRIp, 09RI, 09CRI, 09CARI, 10ARI, and 10ARIn could not indicate the reference solution.

Figure 7 displays EMGM solution evaluations for the DFC data set. 07CRI, 08BRIp, 08BRIm, 09CRI, 09CARI, 09RI, 09BRI, 10QRIp, 10QRIm, 10ARI, 10ARIn, and 11ARInm could not indicate the true number of clusters. 09HI, 10CFn, 11AARInm, 13GRI, and 13AGRI attained their maxima 1 for the right number of clusters. However, 10CFn and 13GRI showed little to no evaluation change over the solutions with number of clusters greater than $k^* = 9$ (low contrast). 10CF attained 0.92 for the right number of clusters.

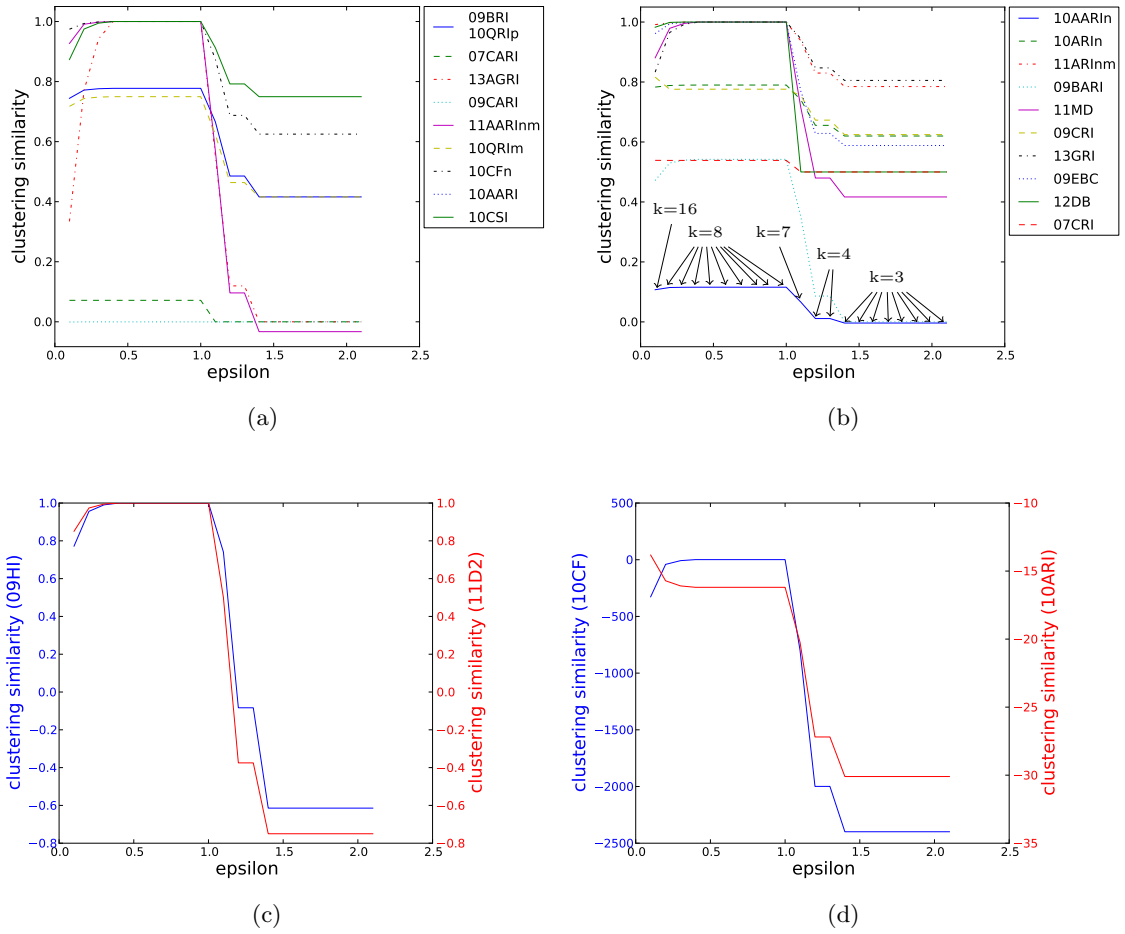


Figure 8: NEHC measure evaluations of SUBCLU solutions for the DNEHC data set.

Figure 8, in which NEHCs are evaluated, shows only the range $\{0.1, 0.2, \ldots, 2.1\}$ of epsilons, as the results from 1.4 to 5.0 are identical. The reference solution has 8 clusters: 4 from data on the plane, 2 from data projected onto the x axis, and 2 from data projected onto the y axis (Figure 4(c)). Figure 8(b) indicates the number of clusters found for each epsilon. SUBCLU generates the reference solution only for the epsilons from 0.4 to 1.0 (we know this by inspection), and most measures yield the highest score in this interval. 07CRI, 09CRI, 10ARI, and 10AARI judged the solution with an epsilon equal to 0.1 to be the best one. Most of the measures identified the correct solutions, but only 09EBC, 09HI, 10CSI, 10CF, 10CFn, 11AARInm, 11MD, 11D2, 13GRI, and 13AGRI attained their maxima 1 for these solutions. 11AARInm and 13AGRI rapidly approached zero for non-optimal epsilons.

In Figure 9, 13GRI and 13AGRI exhibited a steep fall in the evaluations and a peak 1 at the true number of clusters. The DPC data set has only 3 clusters, while the others have 9 (DEHC and DFC) or 8 (DNEHC) clusters. A steeper curve is therefore expected. 07CRI, 09HI, 09BRI, 10QRIp, 10QRIm, 10ARI, 10ARIn, and 11ARInm provided high evaluations

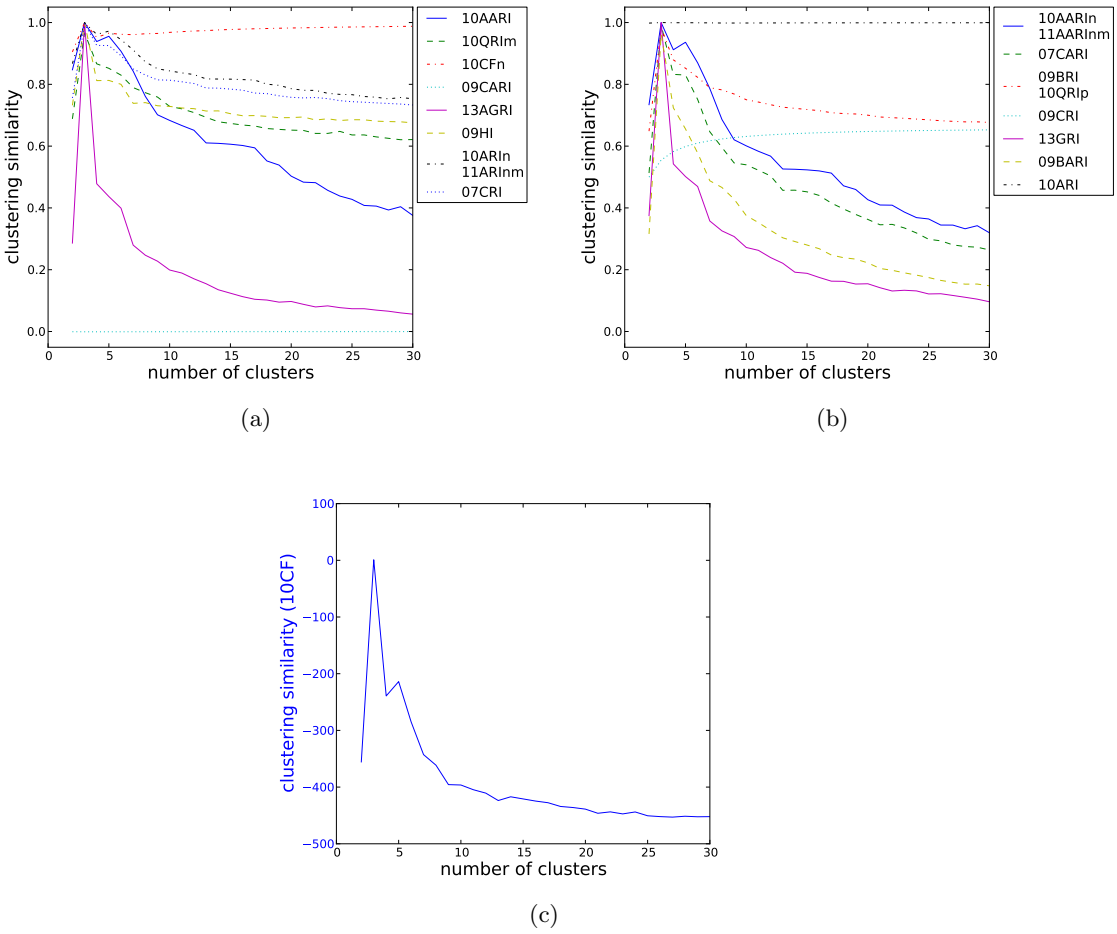


Figure 9: PC measure evaluations of IPCM2 solutions for the DPC data set.

for a wide range of numbers of clusters. Measures 10ARI and 09CARI could not discriminate between the solutions, and 09CRI could not indicate the true number of clusters. 10CFn showed an increasing evaluation for solutions with number of clusters greater than $k = 5$. 10CF indicated the right number of clusters in Fig 9(c), though not evaluating it as the maximum 1 (it was evaluated as 0.92).

Table 3 summarizes the results by indicating with "k*" the measures that identified the reference clustering (discriminant property) and "1" the measures that attained their maxima for the reference solution (maximum property). 09HI, 10CFn, 11AARInm, 13GRI, and 13AGRI are the only measures that displayed the above properties for each scenario. However, 09HI, 10CFn, and 13GRI presented a poor sensitivity to solution variations in most of the cases (e.g., Figures 5(a) and 5(b)), and 10CFn showed an increasing evaluation for progressively worse solutions (Figure 9(a)). 11AARInm and 13AGRI identified the reference solution, attained their maxima 1 for the reference clustering, and were sensitive to the difference in the numbers of clusters in all scenarios.

Measures	EHC	FC^{FCM}	FC^{EMGM}	NEHC	PC
JI	$k^*/1$	-	-	-	-
RI	$k^*/1$	-	-	-	-
ARI	$k^*/1$	-	-	-	-
BC	$k^*/1$	-	-	-	-
03MI	$k^*/1$	$k^*/\cdot$	$k^*/\cdot$	-	-
05MI	$k^*/1$	$k^*/\cdot$	$k^*/\cdot$	-	-
03VI	$k^*/1$	$\cdot/\cdot$	$k^*/\cdot$	-	-
07CRI	$k^*/1$	$k^*/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$	$k^*/\cdot$
07CARI	$k^*/1$	$k^*/\cdot$	$k^*/\cdot$	$\cdot/\cdot$	$k^*/\cdot$
08BRIp	$k^*/1$	$\cdot/\cdot$	$\cdot/\cdot$	-	-
08BRIm	$k^*/1$	$k^*/\cdot$	$\cdot/\cdot$	-	-
09EBC	$k^*/1$	-	-	$k^*/1$	-
09CRI	$\cdot/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$
09CARI	$\cdot/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$
09HI	$k^*/1$	$k^*/1$	$k^*/1$	$k^*/1$	$k^*/1$
09RI	$k^*/1$	$\cdot/\cdot$	$\cdot/\cdot$	-	-
09BRI	$k^*/1$	$k^*/\cdot$	$\cdot/\cdot$	$k^*/\cdot$	$k^*/\cdot$
09BARI	$k^*/1$	$k^*/\cdot$	$k^*/\cdot$	$k^*/\cdot$	$k^*/\cdot$
10QRIp	$k^*/1$	$k^*/\cdot$	$\cdot/\cdot$	$k^*/\cdot$	$k^*/\cdot$
10QRIm	$k^*/1$	$k^*/\cdot$	$\cdot/\cdot$	$k^*/\cdot$	$k^*/\cdot$
10ARI	$k^*/1$	$\cdot/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$
10AARI	$k^*/1$	$k^*/\cdot$	$k^*/\cdot$	$\cdot/\cdot$	$k^*/1$
10ARIn	$k^*/1$	$\cdot/\cdot$	$\cdot/\cdot$	$\cdot/\cdot$	$k^*/1$
10AARIn	$k^*/1$	$k^*/\cdot$	$k^*/\cdot$	$\cdot/\cdot$	$k^*/1$
10CSI	$k^*/1$	-	-	$k^*/1$	-
10CF	$k^*/1$	$k^*/\cdot$	$k^*/\cdot$	$k^*/1$	$k^*/\cdot$
10CFn	$k^*/1$	$k^*/1$	$k^*/1$	$k^*/1$	$k^*/1$
11ARInm	$k^*/1$	$k^*/1$	$\cdot/\cdot$	$\cdot/\cdot$	$k^*/1$
11AARInm	$k^*/1$	$k^*/1$	$k^*/1$	$k^*/1$	$k^*/1$
11MD	$k^*/1$	-	-	$k^*/1$	-
11D2	$k^*/1$	-	-	$k^*/1$	-
13GRI	$k^*/1$	$k^*/1$	$k^*/1$	$k^*/1$	$k^*/1$
13AGRI	$k^*/1$	$k^*/1$	$k^*/1$	$k^*/1$	$k^*/1$
12DB	$k^*/1$	-	-	$\cdot/\cdot$	-

"k^*" means that the measure identified the reference clustering, and "1" means that the measure attained its maximum 1 for the identified reference clustering. A cell with "-" denotes that the measure was not developed for the corresponding clustering type.

Table 3: Maximum and discriminant properties displayed by measures.

7.2 Comparing Randomly Generated Clusterings

The experiment in this section is based on a previously published one (Vinh et al., 2009, 2010) that assessed the ability of proposed EHC measures (based on information theory) to yield a constant baseline for randomly generated solutions. For a particular clustering type (EHC, FC, NEHC, or PC), random model (uniform, beta, unbalanced, or unbalanced-beta), 2-tuple (n, k^*), and $k \in \{2, 3, \ldots, 2k^*\}$, we generated 30 clustering pairs with n objects. Each pair contains a clustering with k clusters (representing an obtained solution) and a clustering with k^* clusters (representing a reference solution). We used four combinations of the number of objects and the true number of clusters: $(n = 25, k^* = 5)$, $(n = 100, k^* = 5)$, $(n = 50, k^* = 10)$, and $(n = 200, k^* = 10)$. The random models used to generate the clusterings depended on the clustering type as follows:

- For EHC, we generated clusterings for both the uniform and unbalanced models. In the uniform model, each object was uniformly assigned to one cluster. In the unbalanced model, each object was assigned to one cluster according to the following distribution: $p_1 \triangleq 0.1/k$ and $p_j \triangleq p_{j-1} + \alpha$ s.t. $\sum_{j=1}^{k} p_j = 1$ (it implies that $\alpha = 1.8/(k(k-1))$), where p_j is the probability of assigning an object to the jth cluster;

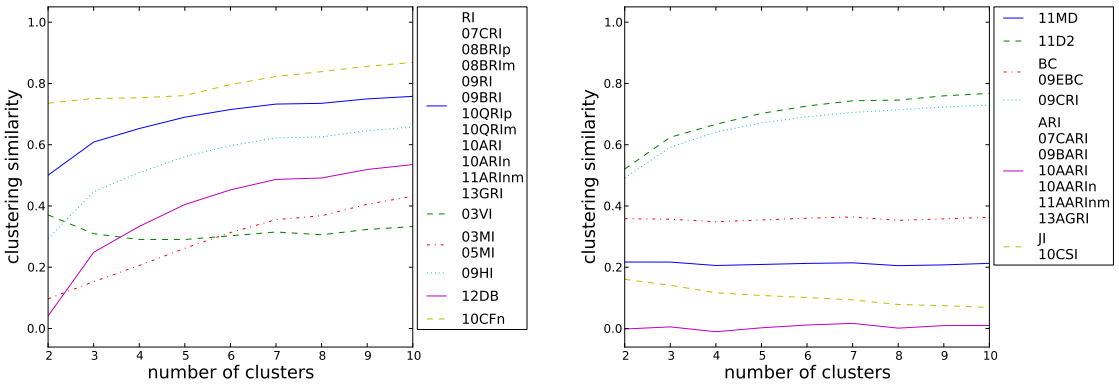
- For FC, we generated clusterings for the uniform, beta, and uniform-beta models. Let X_r^u be a random variable distributed according to the uniform distribution $\mathcal{U}(0, 1)$. For the uniform model, object x_i has a degree of membership to the rth cluster distributed according to $X_r^u/(X_1^u + X_2^u + \cdots + X_k^u)$, where k is the number of clusters. For the beta model, we uniformly draw $r_i \in \mathbb{N}_{1,k}$ for each object x_i to indicate to which cluster x_i probably has the highest degree of membership. Formally, let X_r^b and Y^b be two random variables distributed according to the beta distributions $\mathrm{Be}(1, 5)$ and $\mathrm{Be}(5, 1)$, respectively. Object x_i has a degree of membership to the rth cluster $(r \neq r_i)$ distributed according to $X_r^b/(X_1^b + \cdots + X_{r_i-1}^b + Y^b + X_{r_i+1}^b + \cdots + X_k^b)$ and to the r_ith cluster distributed according to $Y^b/(X_1^b + \cdots + X_{r_i-1}^b + Y^b + X_{r_i+1}^b + \cdots + X_k^b)$. The unbalanced-beta is equal to the beta model except that $r_i \triangleq 1$, such that the first cluster will have most of the membership;

- For NEHC, we generated clusterings for both the uniform and unbalanced models. In the uniform model, each object x_i was uniformly assigned to $k_i \in \mathbb{N}_{1,k}$ clusters, where k_i was uniformly drawn. In the unbalanced model, each object x_i was assigned to $k_i \in \mathbb{N}_{1,k}$ clusters according to the following method. Object x_i is assigned to a cluster according to the distribution p as in the EHC unbalanced model. The distribution p is then adjusted such that the cluster already drawn (say, the jth cluster) will not be selected again for x_i (i.e., $p_j \leftarrow 0$) and normalized to sum 1. The second cluster is randomly selected according to the resulting p. This process is repeated until x_i is assigned to k_i clusters;

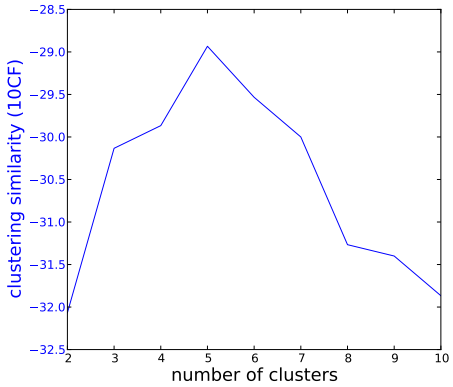
- For PC, we generated clusterings for the uniform, beta, and uniform-beta models. The distributions used are similar to those used for FC. The only difference is the absence of normalizing denominators in their definitions.

Cluster	(EHC, Un)	(FC, UBe)	(NEHC, Un)	(PC, UBe)
1st	2	10.2	22	15.9
2nd	11	10.4	53	16.4
3rd	20	11.4	64	17.0
4th	29	11.7	73	19.0
5th	38	56.2	74	83.3

Table 4: Object-to-cluster membership sums for clustering samples having $n = 100$ objects and $k^* = 5$ clusters.



(a) (b)



(c)

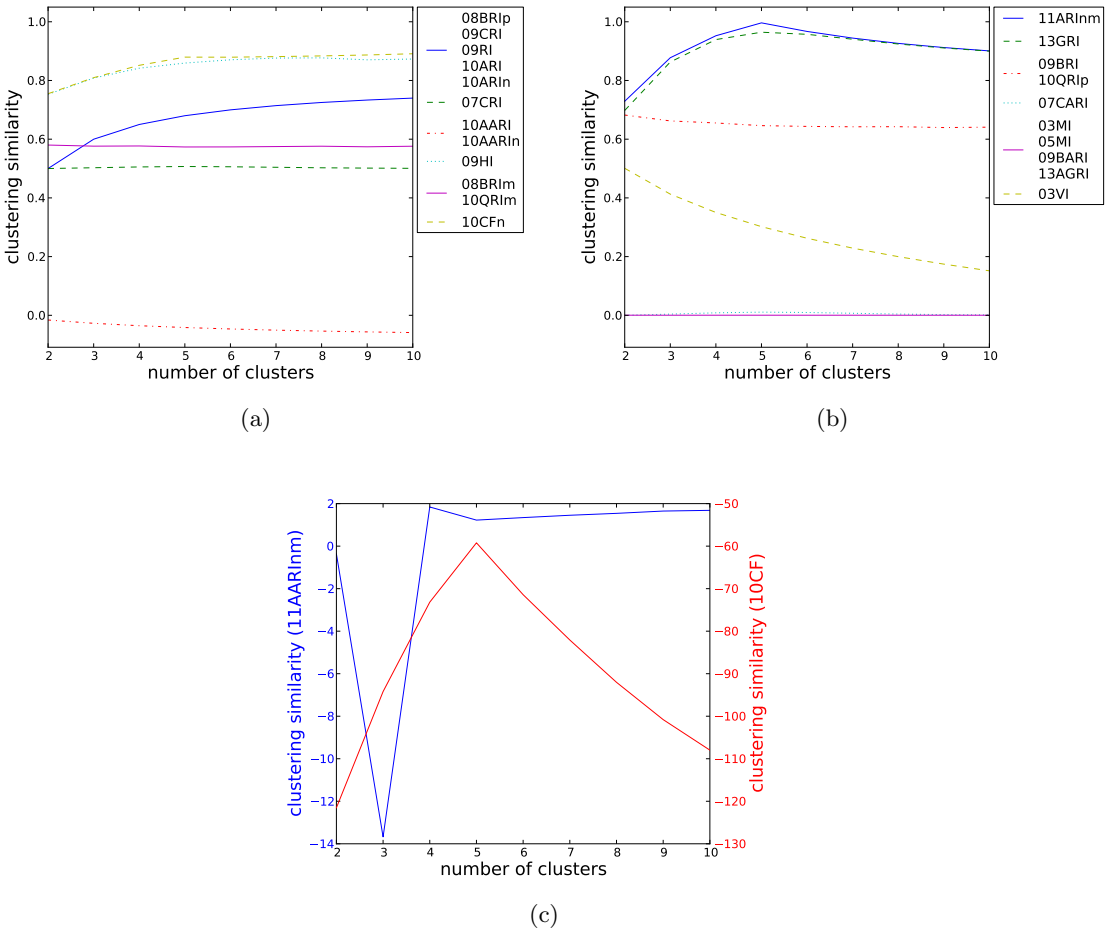
Figure 10: Average evaluations for $(\text{EHC}, \mathcal{U}, n = 25, k^* = 5)$.

We denote a particular experimental setting using a 4-tuple. For example, $(\text{EHC}, \mathcal{U}, n = 25, k^* = 5)$ refers to an EHC set generated according to the uniform model, where each clustering has 25 objects. The solutions of $(\text{EHC}, \mathcal{U}, n = 25, k^* = 5)$ were arranged in 30 EHC pairs for each $k \in \{2, 3, \ldots, 10\}$. Each pair contains an EHC with k clusters and an EHC with k^* clusters. Thus, the set $(\text{EHC}, \mathcal{U}, n = 25, k^* = 5)$ has $30 \cdot 9 = 270$ pairs of clusterings. The measures were then applied to evaluate the similarity between the two clusterings of each EHC pair, and the average evaluation for each $k \in \{2, 3, \ldots, 10\}$ was calculated and plotted in Figure 10. Similarly, Figures 11, 12, and 13 refer to the experimental settings $(\text{FC}, \mathcal{U}, n = 100, k^* = 5)$, $(\text{NEHC}, \mathcal{U}, n = 50, k^* = 10)$, and $(\text{PC}, \text{Be}, n = 200, k^* = 10)$, respectively. The remaining figures are not shown here to avoid cluttering but can be found in the supplementary material: `http://sn.im/25a9h8u`. Those figures will be referred here when appropriate.

Figures 10(a) and 10(b) show that 11 measures exhibited the same averages as RI and that six measures displayed the same averages as ARI, respectively. RI and JI (to a lesser extent) do not show a constant baseline (Hubert and Arabie, 1985; Albatineh et al., 2006), and this behavior is again observed in Figures 10(a) and 10(b). The 13GRI and 13AGRI measures showed the same averages as RI and ARI, respectively, because of their equivalence in the EHC context (Corollaries 4 and 5 in Appendix). 10CF attained a peak at $k = k^*$ clusters in Figure 10(c) for randomly generated clusterings. BC, 09EBC, 11MD, ARI, and the measures with similar values to ARI are the only ones that showed a constant baseline. The others showed a tendency to favor solutions with a high or low numbers of clusters.

Figure 11 shows the results for the experimental setting $(\text{FC}, \mathcal{U}, n = 100, k^* = 5)$. 03MI, 05MI, 07CARI, 09BARI, and 13AGRI displayed a constant baseline close to zero in Figure 11(b). 07CRI, 08BRIm, and 10QRIm (Figure 11(a)) also showed constant baselines, although not close to zero. These three measures were neither formally adjusted for chance nor based on a measure that was. Moreover, 07CRI, 08BRIm, and 10QRIm showed a low variance for a wide range of numbers of clusters in Figure 6. This leads us to suspect that the uniform behavior presented in Figure 11(a) is due to a poor sensitivity to solution variations. 09BRI and 10QRIp exhibited in Figure 11(b) a monotonically decreasing curve with low variation in values, as well as 10AARI and 10AARIn in Figure 11(a). 11AARInm produced values greater than its supposed maximum 1 and showed a counterintuitive behavior in Figure 11(c). 10CF, 11ARInm, and 13GRI showed a peak at $k = k^*$ for randomly generated clusterings.

07CARI, 09BARI, and 13AGRI are the only measures that displayed an approximately constant baseline close to zero in Figure 12, corresponding to the results for $(\text{NEHC}, \mathcal{U}, n = 50, k^* = 10)$. As for 10QRIm in Figure 11(a), the 10QRIp measure had a constant baseline in Figure 12(a) probably due to a low sensitivity in solution discrimination, as it is not adjusted for chance and is based on a measure (RI) known to be biased. The same cannot be said about 13AGRI, as it compares the solutions against a null model and exhibited a strong sensitivity in all experiments in Section 7.1. 10AARI showed in Figure 12(b) values greater than 1 for most solutions. 10CF (Figure 12(e)) and 13GRI (Figure 12(b)) again showed a peak at $k = k^*$ for randomly generated solutions. 10ARI and 11AARInm (Figure 12(d)) produced highly irregular evaluations. 11AARInm produced $-\infty$ (overflow) for $k = 2$ due to near-zero division.



(a) (b)

(c)

Figure 11: Average evaluations for $(\mathrm{FC}, \mathcal{U}, n = 100, k^* = 5)$.

Figure 13 illustrates the results for $(\mathrm{PC}, \mathrm{Be}, n = 200, k^* = 10)$. 09BRI, 09BARI, 10QRIp, 10QRIm, and 13AGRI showed constant baselines, and the constant baselines of 13AGRI and 09BARI were close to zero. 10CF (Figure 13(d)) and 13GRI (Figure 13(b)) again scored random clusterings with $k = k^*$ as better solutions. 10AARI and 11AARInm displayed highly unexpected values (Figure 13(c)).

Table 5 denotes which measures showed the baseline property. The italic n's refer to measures that provided constant baselines in the experiments corresponding to Figures 10, 11, 12, and 13 but not for all the remaining experiments. For example, BC and 09EBC showed unbiased evaluations in Figure 10(b) but not in the experiment $(\mathrm{EHC}, \mathrm{Un}, n = 100, k^* = 5)$ reported in the supplementary material.

Most measures could not provide an unbiased evaluation. They usually tend to favor random solutions with high or low numbers of clusters or show a peak in evaluating random solutions with the same number of clusters as the reference one. This behavior is undesirable, as the compared solutions were independently generated. Only 09BARI and 13AGRI

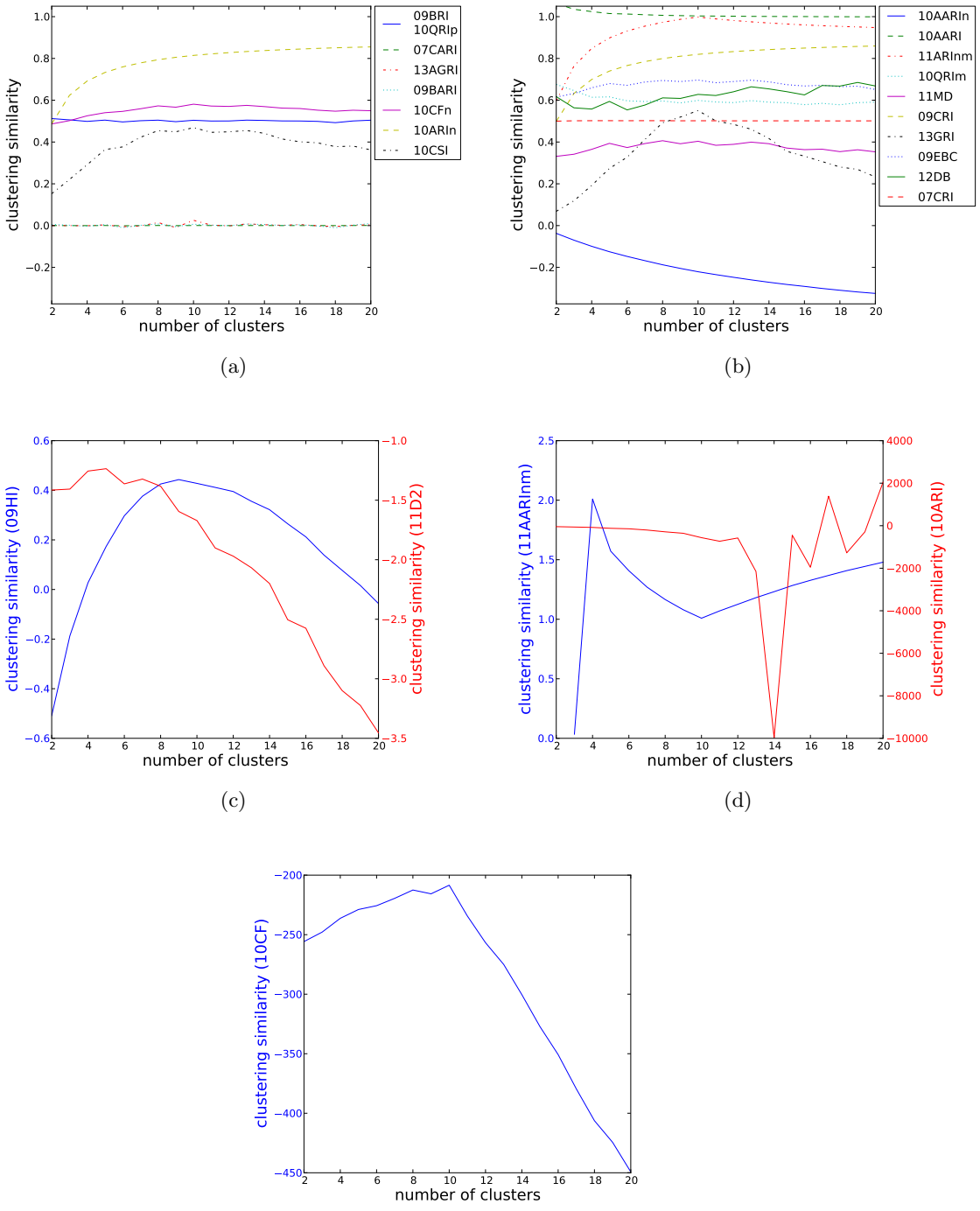


Figure 12: Average evaluations for (NEHC, $\mathcal{U}, n = 50, k^* = 10$).

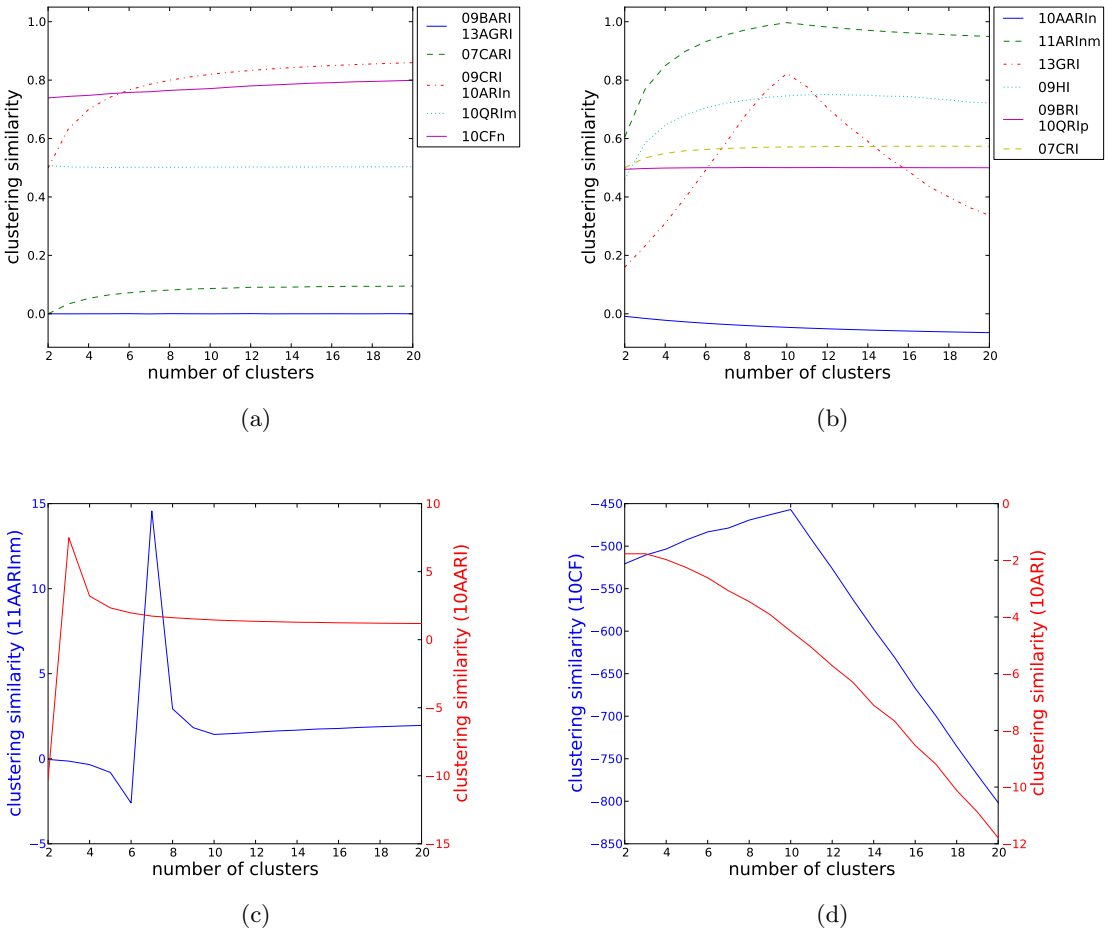


Figure 13: Average evaluations for $(\mathrm{PC}, \mathrm{Be}, n = 200, k^* = 10)$.

presented an approximately constant (and close to zero) baseline in all scenarios. The null model of 13AGRI is clearly violated in each scenario, which suggests that adjusting 13GRI is not just a theoretical adornment but a true correction that makes practical clustering comparisons fairer. Recall that, contrary to 13AGRI, 09BARI did not assign the maximum score 1 to the perfect solutions for all but the EHC scenario in the previous section.

7.3 13AGRI Evaluation Validity for FCs

We applied the k-means and FCM algorithms 30 times for each number of clusters $k \in \{2, 3, \ldots, 20\}$ to the UCI data sets (Newman and Asuncion, 2010) shown in Table 6. 13AGRI evaluated the best clustering (according to the respective algorithm's cost function) for each number of clusters using the known classification as the reference solution; the reference solution is thus an EHC. 13AGRI provides the same evaluation as ARI for k-means solutions since k-means produces EHCs (Corollary 5 in Appendix). FCM is regarded as the fuzzy version of k-means, both search for spherical-like clusters, and FCM tends to k-means when

Measures	EHC	FC	NEHC	PC	Measures	EHC	FC	NEHC	PC
JI	n	-	-	-	09BARI	**y**	**y**	**y**	**y**
RI	n	-	-	-	10QRIp	n	n	*n*	*n*
ARI	**y**	-	-	-	10QRIm	n	*n*	n	*n*
BC	*n*	-	-	-	10ARI	n	n	n	n
03MI	n	**y**	-	-	10AARI	**y**	n	n	n
05MI	n	**y**	-	-	10ARIn	n	n	n	n
03VI	n	n	-	-	10AARIn	**y**	n	n	n
07CRI	n	*n*	**y**	n	10CSI	n	-	n	-
07CARI	**y**	*n*	**y**	n	10CF	n	n	n	n
08BRIp	n	n	-	-	10CFn	n	n	n	n
08BRIm	n	*n*	-	-	11ARInm	n	n	n	n
09EBC	*n*	-	n	-	11AARInm	**y**	n	n	n
09CRI	n	n	n	n	11MD	*n*	-	n	-
09CARI	-	-	-	-	11D2	n	-	n	-
09HI	n	n	n	n	13GRI	n	n	n	n
09RI	n	n	-	-	13AGRI	**y**	**y**	**y**	**y**
09BRI	n	n	*n*	*n*	12DB	n	-	n	-

Table 5: Did the similarity measure display approximately constant baselines?

FCM exponent m approaches 1 (Yu et al., 2004). Thus, their solutions are often similar in the sense that converting an FCM solution into an EHC (by assigning the objects to the clusters for which they have the highest membership degrees) results in a clustering in which the relative assignment of objects is similar to the relative assignment of objects in the solution produced by k-means (i.e., when objects x_i and x_j are assigned to the same cluster in one solution, they are often assigned to the same cluster in the other solution). This section examines whether 13AGRI produces similar evaluations for solutions generated by k-means and FCM. If this is the case, we can be more confident in the validity of 13AGRI FC evaluations since 13AGRI and ARI are equivalent in the EHC domain.

For each data set, Table 7 displays the Pearson correlations between 13AGRI evaluations of the solutions produced by k-means and of the solutions produced by FCM across the number of clusters in $\{2, 3, \ldots, 20\}$. Five correlations were higher than 0.9, and more than a half were higher than 0.7. Figures 14(a) and 14(b) depict 13AGRI evaluations for the data sets on which the correlations attained the three highest and three lowest values, respectively. Figure legends display the corresponding data set, clustering type, and the number of classes in the a priori classification. Because the reference solutions are EHCs, 13AGRI almost always provided higher scores when evaluating EHC solutions than when evaluating FC solutions. The lowest correlations seem to have been obtained in the data sets for which the algorithms could not find good clusterings. For these data sets, the similarity between the found solutions and the reference one mostly fluctuates across the numbers of clusters as (we conjecture) there is no ideal number of clusters at which a peak on the

1. The original data set has 16 objects with missing attributes. We adopted the k-nearest neighbor algorithm with Euclidean distance for imputation (Hastie et al., 1999) and used the resulting data set.

Name	# Objects	# Attributes	# Classes
Breast cancer w. d. (bcw-d)	569	30	2
Breast cancer w. o. (bcw-o)[1]	699	9	2
Synthetic control chart (chart)	600	60	6
Ecoli data set (ecoli)	336	7	7
Glass identification (glass)	214	9	6
Haberman (haberman)	306	3	2
Image segmentation (img)	210	19	7
Ionosphere (ion)	351	34	2
Iris (iris)	150	4	3
Pima indians diabetes (pima)	768	8	2
Connectionist bench (sonar)	208	60	2
SPECT heart (heart)	267	22	2
Vehicle silhouettes (vehicle)	846	18	4
Wine (wine)	178	13	3

Table 6: UCI data sets.

evaluation curve would be found. K-means and FCM produced rather poor solutions for the haberman and sonar data sets according to 13AGRI. 13AGRI evaluations indicate that k-means could uncover some structure in the chart data set because a 13AGRI score (also an ARI score) of 0.5 is a considerable one according to our experience. However, there was not a distinctive solution across the numbers of clusters. 13AGRI indicates the FC solution with three clusters as the most similar to the reference one for the chart data set.

To further investigate the behavior of 13AGRI for the chart solutions, we reduced the chart dimensionality by projecting the 60-dimensional data to the first nine principal components (Jolliffe, 2002) explaining 90% of the variance. We identified two pairs of classes with high degree of overlap (namely, classes *decreasing trend* with *downward shift* and *increasing trend* with *upward shift* (Alcock, 1999)) by projecting the data onto several planes. We joined the classes *decreasing trend* with *downward shift* and *increasing trend* with *upward shift*, resulting in a classification (used as the reference clustering) with four classes. The Pearson correlation between 13AGRI evaluations is now 0.91 using the same experimental configuration as above. Figure 15 shows the evaluations for k-means and FCM solutions. 13AGRI provided high evaluations for k-means solutions with three and four clusters, while 13AGRI suggests that the best FCM solution is the one with three clusters.

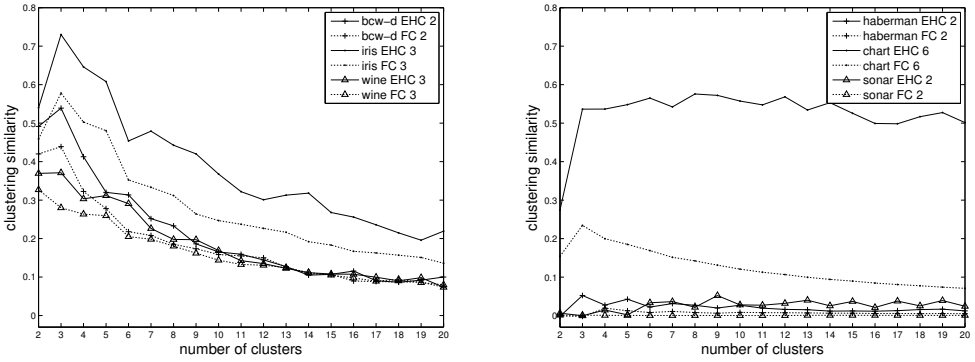
Results indicate that 13AGRI when applied to FCs behaves similarly to 13AGRI (i.e., ARI) when applied to EHCs, particularly when the solutions uncover some data set structure. Considering that ARI is one of the most trusted similarity measures, the results corroborate the 13AGRI evaluation validity for FCs.

7.4 Clustering Stability Assessment

We applied EMGM to subsamples of the top five data sets from the previous section (i.e., bcw-d, iris, wine, bcw-o, and img) 100 times for each number of clusters $k \in \{2, \ldots, 20\}$, generating 100 Gaussian mixtures for each number of clusters and data set; these Gaussian

bcw-d	iris	wine	bcw-o	img	ecoli	ion
0.99	0.99	0.98	0.98	0.91	0.89	0.83
vehicle	glass	pima	heart	haberman	chart	sonar
0.75	0.70	0.69	0.60	0.23	0.02	-0.45

Table 7: Correlation between 13AGRI evaluations of hard exclusive and fuzzy clusterings.



(a) (b)

Figure 14: 13AGRI evaluations that exhibited the three highest (a) and the three lowest correlations (b).

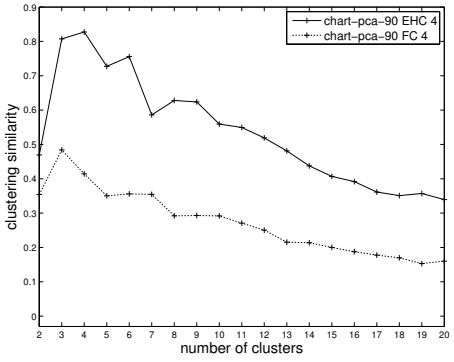


Figure 15: 13AGRI evaluations for the processed chart data set.

bcw-d	bcw-o	wine	iris	img
0.94	0.90	0.85	0.67	0.59

Table 8: Correlation between 13AGRI evaluation and stability statistic.

mixtures are different explanations for the phenomenon that produced the data set. We calculated a probabilistic clustering U (also known as FC) of the whole data set for each Gaussian mixture such that $U_{r,i}$ is the probability of x_i belonging to the rth cluster (i.e., to the rth Gaussian mixture component). 13AGRI compared each of the $\binom{100}{2}$ probabilistic clustering pairs for each number of clusters and data set, and the average was taken as the stability statistic (the less diverse the solution set, the higher the stability statistic) for the corresponding number of clusters and data set. Subsamples were generated by randomly selecting 80% of the data set objects, without replacement, as in (Monti et al., 2003). Algorithm 2 describes how stability assessment can be used to estimate the number of clusters and to select a promising clustering of a set of solutions.

Algorithm 2 Stability assessment

Require: Data set X.

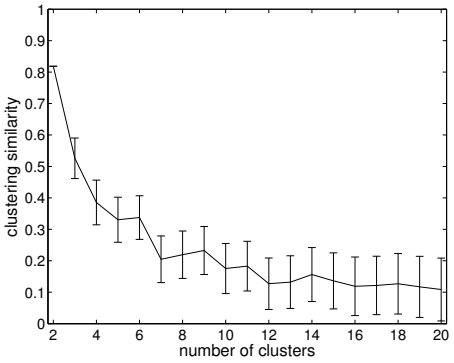
1: **for** $i \in \{1, 2, \ldots, 100\}$ **do**
2: $S_i \leftarrow$ Randomly draw 80% of the objects from X, without reposition.
3: **end for**
4: **for** $k \in \{2, 3, \ldots, 20\}$ **do**
5: **for** $i \in \{1, 2, \ldots, 100\}$ **do**
6: Apply EMGM to S_i finding a Gaussian mixture with k components.
7: $U^i \leftarrow$ Calculate the probabilistic clustering of the whole data set using the found Gaussian mixture.
8: **end for**
9: $t_k \leftarrow \sum_{i<j} 13\text{AGRI}(U^i, U^j)/\binom{100}{2}$ {stability statistic}
10: $V^k \leftarrow \text{argmax}_{U^i}\{\sum_{j \neq i} 13\text{AGRI}(U^i, U^j)\}$ {clustering set prototype}
11: **end for**
12: $k' \leftarrow \text{argmax}_{k \in \{2,\ldots,20\}}\{t_k\}$ {estimated number of clusters}
13: $U' \leftarrow V^{k'}$; {estimated best clustering}

Table 8 shows the Pearson correlations between stability statistic (defined by Step 9) values and 13AGRI evaluations (similarity between prototype V^k, Step 10, and the reference clustering) for different number of clusters. The high correlations indicate that the stability statistic, which can be used in real scenarios, approximately follows the 13AGRI evaluation that depends on a reference solution.

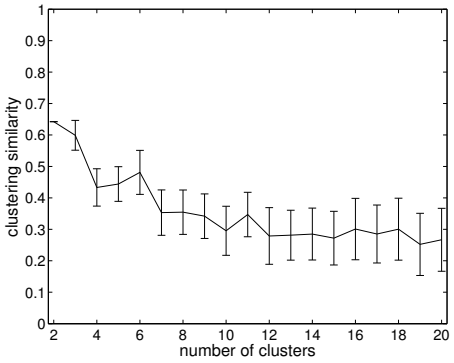
Figure 16 depicts 13AGRI evaluation for each clustering set prototype (Step 10 in Algorithm 2) and data set. We generated the error bar for a given $k \in \{2, \ldots, 20\}$ and data set as follows. Let t_k be the stability statistic for the set of clusterings with k clusters each (Step 9). Error bar was calculated to take 0 for the more stable clustering set (highest stability statistic) and 0.1 for the least stable clustering set, for visualization purposes. Thus, the

error bar value corresponding to the set of clusterings with k clusters is

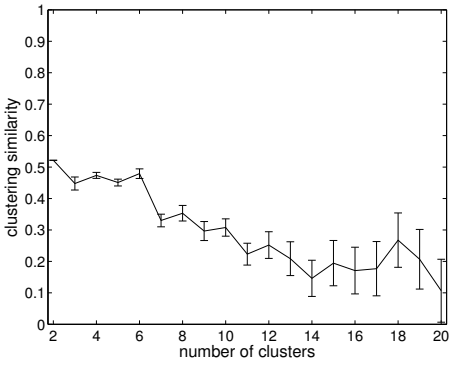
$$\frac{t_k - \min\{t_2, t_3, \ldots, t_{20}\}}{\max\{t_2, t_3, \ldots, t_{20}\} - \min\{t_2, t_3, \ldots, t_{20}\}} \times 0.1.$$



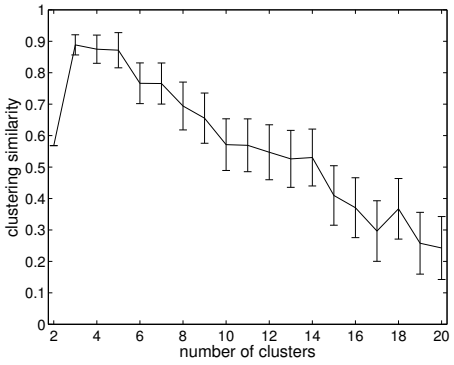
(a) Data set: bcw-d ($k^* = 2$).



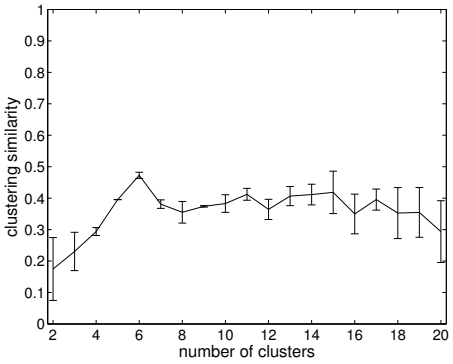
(b) Data set: bcw-o ($k^* = 2$).



(c) Data set: wine ($k^* = 3$).



(d) Data set: iris ($k^* = 3$).



(e) Data set: img ($k^* = 7$).

Figure 16: 13AGRI evaluations with error bars indicating clustering set instability.

Stability statistic precisely estimated the correct number of clusters for bcw-d (Figure 16(a)) and bcw-o (Figure 16(b)) data sets. The top two stable clustering sets in iris are the

ones with two and three clusters. Iris data set is classified in three classes (namely, *setosa*, *versicolour*, and *virginica*). However, it is well-known that the *versicolour* and *virginica* classes have a high degree of overlap and are frequently considered a single cluster (Wu and Yang, 2005), which corroborate the validity of the stability statistic. Although not being able to indicate the exact number of clusters, the lowest instability values for wine and img are around the correct number of clusters. In general, the near the number of clusters of the clustering set to the ideal one, the more stable the clustering set tends to be. These good preliminary results demonstrate that 13AGRI deserves further investigations on its applicability to the estimation of the number of clusters for FCs.

8. Discussion

Sections 7.1 and 7.2 empirically explored the four measure properties proposed in Section 2. Section 7.1 investigated the maximum, discriminant, and contrast properties by applying the measures to gradually different solutions. The hypothesis was that the similarity between the found clustering and the reference one is highly correlated to the difference in the number of clusters (epsilons in the case of SUBCLU) between the compared solutions, given that the solutions are produced by clustering algorithms capable of finding the ideal solution. One can understand the difference between the number of clusters given to the algorithm and the number of clusters of the reference solution as how far the domain of solutions of the corresponding algorithm is to the reference clustering. It is expected that a good measure should translate that difference in terms of evaluations. Section 7.1 showed that several of the measures did not follow the above hypothesis or did so in a very loose way, showing almost flat evaluations over the number of clusters. Moreover, several measures could not discriminate the best solution (03VI, 07CRI, 07CARI, 08BRIp, 08BRIm, 09CRI, 09CARI, 09RI, 09BRI, 10QRIp, 10QRIm, 10ARI, 10AARI, 10ARIn, 10AARIn, 11ARInm, and 12DB) for at least one of the clustering domains considered. We believe that this result by itself is enough for considering those measures unsuitable for the clustering domains they have failed. Section 7.1 concluded that 03MI, 05MI, 09BARI, and 10CF (beside the ones that have failed for the discriminant property) did not show the maximum property, and several measures were poorly sensitive to different solution qualities (poor contrast).

The baseline property was investigated in Section 7.2. In particular, we aimed to find out what measures were able to perform unbiased evaluations over different numbers of clusters. We concluded that only 09BARI and 13AGRI showed the baseline property for every clustering domain. By correcting 13GRI for chance, we were striving to build a measure that can capture the similarity between two solutions irrespectively to their numbers of clusters. We thus implicitly assumed that the number of clusters is not per se an indication of the similarity between clusterings (Section 7.2) but only a factor that delineates the domain of solutions (Section 7.1).

The correction-for-chance property implemented for 13AGRI, and that other measures displayed in Section 7.2 for certain scenarios, can also be understood as a way to stretch out the measure such that its useful range lies between the constant baseline and the maximum. As a matter of fact, one is not usually interested in very poor solutions (i.e., the ones that are far from the reference) (Meila, 2012), and those would receive negative or close to zero evaluations by 13AGRI and other adjusted measures. The correction-for-

chance thus increases the interpretability of the results by stressing what one should expect from clusterings whose evaluations lie below, close to, or above the baseline.

9. Conclusions

This paper discussed the importance of similarity measures in evaluating clustering algorithms, consensus clustering, clustering stability assessment, and quantifying information loss. These and other applications led to a recent interest in measures (especially pair-based ones) capable of comparing more general clusterings than the exclusive hard ones (usual partitions of an object set). We provided an overview of 28 measures proposed in the past 10 years for this task and discussed some of their issues. We showed that several of these measures do not attain the maximum whenever two equivalent solutions are compared and that most measures are biased toward clusterings with certain numbers of clusters. Moreover, several of the discussed measures are based on equations that were originally developed specifically for and by assuming the exclusive hard domain. Some measures thus exhibited unexpected behavior in experiments involving more general scenarios.

We proposed the 13FRI measure that can be used to compare fuzzy/probabilistic and exclusive hard clusterings. Based on a null model we proposed, according to which clusterings are generated, and following the framework employed by Hubert and Arabie (1985) to adjust the Rand index, 13AFRI was proposed as a corrected-for-chance version of 13FRI. We then extended 13FRI and 13AFRI to handle more general clusterings, namely possibilistic clusterings (including exclusive hard, fuzzy/probabilistic, and non-exclusive hard clusterings), yielding 13GRI and 13AGRI, respectively. The computational complexity analysis showed that our measures are practical.

In the first experiment involving four clustering algorithms of different natures, we observed that some measures could not identify the best solutions, and that several could not provide a fine-grained evaluation across the range of the numbers of clusters, whereas 13AGRI always attained its maximum 1 for the true number of clusters and displayed a steep, discriminative evaluation curve with a clear peak at the true number of clusters for each data set. We assessed the capability of the measures to provide an unbiased evaluation for randomly generated solutions with different numbers of clusters in the second experiment. A fair measure should assign a uniform evaluation across the range of the numbers of clusters, as each generated solution is independent of the reference one (Vinh et al., 2010). This is the case of the well-known adjusted Rand index (ARI) (Hubert and Arabie, 1985) for the exclusive hard domain. Only 13AGRI and 09BARI (Brouwer, 2009) (a recently proposed measure) displayed such an evaluation for all considered scenarios, which include the exclusive hard context; however, 09BARI could not attain its maximum 1 at the true number of clusters for all but the hard exclusive domain in the first experiment. The other measures exhibited a preference for certain solutions, which is attributable solely to their evaluation mechanisms. While the randomness model for 13AGRI incorporates some assumptions about the clusterings, those generated in our experiments clearly do not follow such a requirement. Even so, 13AGRI could provide uniform evaluations close to zero in the experiments with randomly generated solutions.

Two more experiments involving 14 real data sets and the algorithms k-means (MacQueen, 1967), fuzzy c-means (FCM) (Bezdek, 1981), and expectation maximization for

Gaussian mixtures (Dempster et al., 1977) were performed to assess the validity of 13AGRI evaluations in the fuzzy domain, arguably the most important domain after the exclusive hard one, and to investigate 13AGRI's applicability to the estimation of the number of clusters without (of course) any knowledge about the true data structure. We argue that the evaluations of the solutions produced by k-means and FCM for the same data set should be similar, and this behavior presented by 13AGRI is even more important for its validity because 13AGRI and the trusted ARI measures are equivalent when applied to solutions generated by k-means. The stability statistic based on 13AGRI defined in our last experiment showed good results indicating that 13AGRI can also be successfully applied to the estimation of the number of clusters in the probabilistic domain.

We proved that 13AGRI and ARI are equivalent in the exclusive hard domain. This is reassuring because (i) ARI is one of the most trusted similarity measures (Steinley, 2004; Albatineh et al., 2006), and (ii) the null model of 13AGRI was developed for general possibilistic clusterings (including exclusive hard clusterings as a special case). As future work, we think that 13AGRI deserves a further investigation on its conceptual properties, specially those generally taken as useful for similarity measures for clustering, such as cluster homogeneity sensibility, cluster completeness, and metric axioms compliance (Meila, 2007; Amigó et al., 2009).

Acknowledgments

We thank the editor and the anonymous reviewers for their constructive comments. This work was financially supported by the Brazilian agencies CNPq (#304137/2013-8) and FAPESP (#2009/17469-6 & #2013/18698-4).

Appendix A.

Proposition 1 *Let* U *and* V *be two FCs such that* $13\text{FRI}(\mathrm{U},\mathrm{V}) = 0$*,* $n > 1$*, and* $1 \leq k_{\mathrm{U}}, k_{\mathrm{V}} \leq n$*. It implies that* U *and* V *are EHCs and that* $k_{\mathrm{U}} = 1$ *and* $k_{\mathrm{V}} = n$ *or* $k_{\mathrm{U}} = n$ *and* $k_{\mathrm{V}} = 1$*, which unambiguously determine* U *and* V*.*

Proof Realize from Equations (12) that $\sum_{r=1}^{k_{\mathrm{U}}} \mathrm{U}_{r,l} = 1\ \forall l$ implies $\mathrm{S}^{\mathrm{U}}_{i,j} = 1 - \mathrm{J}^{\mathrm{U}}_{i,j}$. To have $13\text{FRI}(\mathrm{U},\mathrm{V}) = 0$, it must be the case that $\dot{a} = \dot{d} = 0$ (Equation 14), which implies that $\min\{\mathrm{J}^{\mathrm{U}}_{i,j}, \mathrm{J}^{\mathrm{V}}_{i,j}\} = \min\{1 - \mathrm{J}^{\mathrm{U}}_{i,j}, 1 - \mathrm{J}^{\mathrm{V}}_{i,j}\} = 0\ \forall i < j$ (Equations 13a and 13d). Hence, $\mathrm{J}^{\mathrm{U}}_{i,j}, \mathrm{J}^{\mathrm{V}}_{i,j} \in \{0,1\}$ and $\mathrm{J}^{\mathrm{U}}_{i,j} \neq \mathrm{J}^{\mathrm{V}}_{i,j}$ for all $i < j$.

We first prove by contradiction that U cannot have a column i and a row r for which $\mathrm{U}_{r,i} \in (0,1)$ (the same holds for V). Assuming that the ith column of U has $\mathrm{U}_{r,i} \in (0,1)$ for an $r \in \mathbb{N}_{1,k_{\mathrm{U}}}$, we have $k_{\mathrm{U}} > 1$ and at least two elements of $\mathrm{U}_{:,i}$ have values in the open interval $(0,1)$ because $\sum_{t=1}^{k_{\mathrm{U}}} \mathrm{U}_{t,i} = 1$. Without loss of generality, assume that $i = 1$ (the columns of U and V can always be simultaneously permuted without changing the measure). We know that $\mathrm{U}^{\mathrm{T}}_{:,1}\mathrm{U}_{:,j} = \mathrm{J}^{\mathrm{U}}_{1,j} = 0\ \forall j \in \mathbb{N}_{2,n}$ because $\mathrm{U}^{\mathrm{T}}_{:,1}\mathrm{U}_{:,j}$ cannot yield 1. Thus, $\mathrm{J}^{\mathrm{V}}_{1,j} = 1\ \forall j \in \mathbb{N}_{2,n}$. This implies that the columns of V are all identical and each one has the element 1, resulting in $k_{\mathrm{V}} = 1$ because of the constraint $\sum_{j=1}^{n} \mathrm{V}_{t,j} > 0\ \forall t$. We thus have $\mathrm{J}^{\mathrm{V}}_{i_1,j_1} = 1\ \forall i_1 < j_1$ and $\mathrm{J}^{\mathrm{U}}_{i_2,j_2} = 0\ \forall i_2 < j_2$. The last equality only holds with constraint

$\sum_{j=1}^{n} \mathrm{U}_{t,j} > 0 \ \forall t$ if each row of U has exactly one value greater than zero. The property $\sum_{t=1}^{k_\mathrm{U}} \mathrm{U}_{t,j} = 1 \ \forall j$ of FCs and the assumption $k_\mathrm{U} \leq n$ then require each column of U to have exactly one value greater than zero (and to have $k_\mathrm{U} = n$ rows), which is the value 1. This violates the assumption that $\mathrm{U}_{r,i} \in (0,1)$, which implies that U (and V) must be a matrix with only zeros and ones.

Suppose $n = 2$. If columns 1 and 2 of U are identical, columns 1 and 2 of V are different because we have already proven that $\mathrm{J}^\mathrm{U}_{i,j} \neq \mathrm{J}^\mathrm{V}_{i,j}$. This only can happen for $k_\mathrm{U} = 1$ and $k_\mathrm{V} = 2$ (remember the properties of an FC). Now, suppose that $n > 2$ and, without loss of generality, that $\mathrm{U}_{:,1}$ and $\mathrm{U}_{:,2}$ are identical and that $\mathrm{V}_{:,1}$ and $\mathrm{V}_{:,2}$ are different. If a column $i > 2$ of U differs from columns 1 and 2 of U, we conclude that columns 1 and 2 of V are equal to column i of V. However, this implies that columns 1 and 2 of V are equal, and, as we known, they are not. Consequently, all columns of U must be identical and all columns of V must be different. This can only happen for $k_\mathrm{U} = 1$ and $k_\mathrm{V} = n$, which proves the proposition. ■

Proposition 2 *Given two EHCs* U *and* V*, we have* $13\mathrm{FRI}(\mathrm{U},\mathrm{V}) = \mathrm{RI}(\mathrm{U},\mathrm{V})$.

Proof Realize that $\dot{a}$, $\dot{b}$, $\dot{c}$, and $\dot{d}$ (Equations 13) are equivalent to a, b, c, and d (Equations 7) by assigning the values 0 and 1 to $\mathrm{J}^\mathrm{U}_{i,j}$ and $\mathrm{J}^\mathrm{V}_{i,j}$. ■

Proposition 3 *Given two EHCs* U *and* V*, we have* $13\mathrm{AFRI}(\mathrm{U},\mathrm{V}) = \mathrm{ARI}(\mathrm{U},\mathrm{V})$.

Proof Both ARI and 13AFRI use the framework of Equation (15). The expectation of ARI given U and V is $\mathrm{E}[\mathrm{ARI}]_{\mathrm{U},\mathrm{V}} = (\mathrm{E}[a]_{\mathrm{U},\mathrm{V}} + \mathrm{E}[d]_{\mathrm{U},\mathrm{V}})/(a+b+c+d)$ (Hubert and Arabie, 1985). We must therefore only show that $\mathrm{E}[a]_{\mathrm{U},\mathrm{V}} = \mathrm{E}[\dot{a}]_{\mathrm{U},\mathrm{V}}$ and $\mathrm{E}[d]_{\mathrm{U},\mathrm{V}} = \mathrm{E}[\dot{d}]_{\mathrm{U},\mathrm{V}}$, since $a = \dot{a}$, $b = \dot{b}$, $c = \dot{c}$, and $d = \dot{d}$ by Proposition 2. Let $\mathrm{J}^\mathrm{U} = \mathrm{U}^\mathrm{T}\mathrm{U}$, $\mathrm{J}^\mathrm{V} = \mathrm{V}^\mathrm{T}\mathrm{V}$, and $\mathrm{N} = \mathrm{U}\mathrm{V}^\mathrm{T}$. Because U and V are EHCs, we can rewrite $\min\{\mathrm{J}^\mathrm{U}_{i,j}, \mathrm{J}^\mathrm{V}_{i,j}\} = \mathrm{J}^\mathrm{U}_{i,j}\mathrm{J}^\mathrm{V}_{i,j}$. Both $\sum_{i<j} \mathrm{J}^\mathrm{U}_{i,j}$ and $\sum_{r=1}^{k_\mathrm{U}} \binom{\mathrm{N}_{r,+}}{2}$ count the number of unordered object pairs in the same cluster in U. We thus have

$$
\begin{aligned}
\mathrm{E}[\dot{a}]_{\mathrm{U},\mathrm{V}} &= \frac{2}{n(n-1)} \sum_{i_1<j_1} \mathrm{J}^\mathrm{U}_{i_1,j_1} \sum_{i_2<j_2} \mathrm{J}^\mathrm{V}_{i_2,j_2} \\
&= \sum_{r=1}^{k_\mathrm{U}} \binom{\mathrm{N}_{r,+}}{2} \sum_{t=1}^{k_\mathrm{V}} \binom{\mathrm{N}_{+,t}}{2} \Big/ \binom{n}{2} \\
&= \mathrm{E}[a]_{\mathrm{U},\mathrm{V}} \text{ (Equation (2) in (Hubert and Arabie, 1985)).}
\end{aligned}
$$

Because $\mathrm{J}^{\mathrm{U}}_{i,j} = 1 - \mathrm{S}^{\mathrm{U}}_{i,j}$ for EHCs, we have

$$\begin{aligned}
\mathrm{E}[\dot{d}]_{\mathrm{U,V}} &= \frac{2}{n(n-1)} \sum_{i_1<j_1} \sum_{i_2<j_2} (1 - \mathrm{J}^{\mathrm{U}}_{i_1,j_1})(1 - \mathrm{J}^{\mathrm{V}}_{i_2,j_2}) \\
&= \binom{n}{2} - \sum_{i_1<j_1} \mathrm{J}^{\mathrm{U}}_{i_1,j_1} - \sum_{i_2<j_2} \mathrm{J}^{\mathrm{V}}_{i_2,j_2} \\
&+ \sum_{i_1<j_1} \sum_{i_2<j_2} \mathrm{J}^{\mathrm{U}}_{i_1,j_1} \mathrm{J}^{\mathrm{V}}_{i_2,j_2} \Big/ \binom{n}{2} \\
&= \binom{n}{2} - \sum_{r=1}^{k_{\mathrm{U}}} \binom{\mathrm{N}_{r,+}}{2} - \sum_{t=1}^{k_{\mathrm{V}}} \binom{\mathrm{N}_{+,t}}{2} \\
&+ \sum_{r=1}^{k_{\mathrm{U}}} \binom{\mathrm{N}_{r,+}}{2} \sum_{t=1}^{k_{\mathrm{V}}} \binom{\mathrm{N}_{+,t}}{2} \Big/ \binom{n}{2} \\
&= \mathrm{E}[d]_{\mathrm{U,V}} \text{ (Equation (3) in (Hubert and Arabie, 1985) multiplied by } \binom{n}{2} \text{ and} \\
&\text{then subtracted by } \mathrm{E}[a]_{\mathrm{U,V}}).
\end{aligned}$$

■

Proposition 4 *Given two PCs* U *and* V*, we have* $\dot{a} + \dot{b} + \dot{c} + \dot{d} = \sum_{i<j} \min\{\mathrm{T}^{\mathrm{U}}_{i,j}, \mathrm{T}^{\mathrm{V}}_{i,j}\}$.

Proof Let

$$\begin{aligned}
\dot{a}_{i,j} &\triangleq \min\{\mathrm{J}^{\mathrm{U}}_{i,j}, \mathrm{J}^{\mathrm{V}}_{i,j}\}, \\
\dot{b}_{i,j} &\triangleq \min\{\mathrm{J}^{\mathrm{U}}_{i,j} - \min\{\mathrm{J}^{\mathrm{U}}_{i,j}, \mathrm{J}^{\mathrm{V}}_{i,j}\}, \mathrm{S}^{\mathrm{V}}_{i,j} - \min\{\mathrm{S}^{\mathrm{U}}_{i,j}, \mathrm{S}^{\mathrm{V}}_{i,j}\}\}, \\
\dot{c}_{i,j} &\triangleq \min\{\mathrm{J}^{\mathrm{V}}_{i,j} - \min\{\mathrm{J}^{\mathrm{U}}_{i,j}, \mathrm{J}^{\mathrm{V}}_{i,j}\}, \mathrm{S}^{\mathrm{U}}_{i,j} - \min\{\mathrm{S}^{\mathrm{U}}_{i,j}, \mathrm{S}^{\mathrm{V}}_{i,j}\}\}, \text{ and} \\
\dot{d}_{i,j} &\triangleq \min\{\mathrm{S}^{\mathrm{U}}_{i,j}, \mathrm{S}^{\mathrm{V}}_{i,j}\}.
\end{aligned}$$

We prove the proposition by showing that

$$\dot{a}_{i,j} + \dot{b}_{i,j} + \dot{c}_{i,j} + \dot{d}_{i,j} = \min\{\mathrm{T}^{\mathrm{U}}_{i,j}, \mathrm{T}^{\mathrm{V}}_{i,j}\}. \tag{25}$$

Table 9 shows the six rank combinations between the values of the pairs $(\mathrm{J}^{\mathrm{U}}_{i,j}, \mathrm{J}^{\mathrm{V}}_{i,j})$, $(\mathrm{S}^{\mathrm{U}}_{i,j}, \mathrm{S}^{\mathrm{V}}_{i,j})$, and $(\mathrm{T}^{\mathrm{U}}_{i,j}, \mathrm{T}^{\mathrm{V}}_{i,j})$, covering all possible scenarios. Equation (25) is true for each scenario. For conciseness, let us show the proof for Combinations 1 and 3 only.

Assuming Combination 1, we have $\dot{a}_{i,j} = \mathrm{J}^{\mathrm{V}}_{i,j}$, $\dot{b}_{i,j} = 0$, $\dot{c}_{i,j} = 0$, and $\dot{d}_{i,j} = \mathrm{S}^{\mathrm{V}}_{i,j}$, and Equation (25) is true. Assuming Combination 3, we have $\dot{a}_{i,j} = \mathrm{J}^{\mathrm{V}}_{i,j}$, $\dot{b}_{i,j} = \min\{\mathrm{J}^{\mathrm{U}}_{i,j} - \mathrm{J}^{\mathrm{V}}_{i,j}, \mathrm{S}^{\mathrm{V}}_{i,j} - \mathrm{S}^{\mathrm{U}}_{i,j}\}$, $\dot{c}_{i,j} = 0$, and $\dot{d}_{i,j} = \mathrm{S}^{\mathrm{U}}_{i,j}$. Note that $\mathrm{T}^{\mathrm{U}}_{i,j} < \mathrm{T}^{\mathrm{V}}_{i,j} \Rightarrow \mathrm{J}^{\mathrm{U}}_{i,j} + \mathrm{S}^{\mathrm{U}}_{i,j} < \mathrm{J}^{\mathrm{V}}_{i,j} + \mathrm{S}^{\mathrm{V}}_{i,j} \Rightarrow \mathrm{J}^{\mathrm{U}}_{i,j} - \mathrm{J}^{\mathrm{V}}_{i,j} < \mathrm{S}^{\mathrm{V}}_{i,j} - \mathrm{S}^{\mathrm{U}}_{i,j}$. Thus, $\dot{b}_{i,j} = \mathrm{J}^{\mathrm{U}}_{i,j} - \mathrm{J}^{\mathrm{V}}_{i,j}$, and Equation (25) is true. ■

#	$(\mathrm{J}^{\mathrm{U}}_{i,j}, \mathrm{J}^{\mathrm{V}}_{i,j})$	$(\mathrm{S}^{\mathrm{U}}_{i,j}, \mathrm{S}^{\mathrm{V}}_{i,j})$	$(\mathrm{T}^{\mathrm{U}}_{i,j}, \mathrm{T}^{\mathrm{V}}_{i,j})$
1	$\mathrm{J}^{\mathrm{U}}_{i,j} \geq \mathrm{J}^{\mathrm{V}}_{i,j}$	$\mathrm{S}^{\mathrm{U}}_{i,j} \geq \mathrm{S}^{\mathrm{V}}_{i,j}$	$\mathrm{T}^{\mathrm{U}}_{i,j} \geq \mathrm{T}^{\mathrm{V}}_{i,j}$
2	$\mathrm{J}^{\mathrm{U}}_{i,j} \geq \mathrm{J}^{\mathrm{V}}_{i,j}$	$\mathrm{S}^{\mathrm{U}}_{i,j} < \mathrm{S}^{\mathrm{V}}_{i,j}$	$\mathrm{T}^{\mathrm{U}}_{i,j} \geq \mathrm{T}^{\mathrm{V}}_{i,j}$
3	$\mathrm{J}^{\mathrm{U}}_{i,j} \geq \mathrm{J}^{\mathrm{V}}_{i,j}$	$\mathrm{S}^{\mathrm{U}}_{i,j} < \mathrm{S}^{\mathrm{V}}_{i,j}$	$\mathrm{T}^{\mathrm{U}}_{i,j} < \mathrm{T}^{\mathrm{V}}_{i,j}$
4	$\mathrm{J}^{\mathrm{U}}_{i,j} < \mathrm{J}^{\mathrm{V}}_{i,j}$	$\mathrm{S}^{\mathrm{U}}_{i,j} \geq \mathrm{S}^{\mathrm{V}}_{i,j}$	$\mathrm{T}^{\mathrm{U}}_{i,j} \geq \mathrm{T}^{\mathrm{V}}_{i,j}$
5	$\mathrm{J}^{\mathrm{U}}_{i,j} < \mathrm{J}^{\mathrm{V}}_{i,j}$	$\mathrm{S}^{\mathrm{U}}_{i,j} \geq \mathrm{S}^{\mathrm{V}}_{i,j}$	$\mathrm{T}^{\mathrm{U}}_{i,j} < \mathrm{T}^{\mathrm{V}}_{i,j}$
6	$\mathrm{J}^{\mathrm{U}}_{i,j} < \mathrm{J}^{\mathrm{V}}_{i,j}$	$\mathrm{S}^{\mathrm{U}}_{i,j} < \mathrm{S}^{\mathrm{V}}_{i,j}$	$\mathrm{T}^{\mathrm{U}}_{i,j} < \mathrm{T}^{\mathrm{V}}_{i,j}$

Table 9: Rank combinations.

Corollary 1 *If* U *and* V *are two FCs with* n *columns each, we have* $\mathrm{T}^{\mathrm{U}}_{i,j} = \mathrm{T}^{\mathrm{V}}_{i,j} = 1$ *and the sum* $\dot{a} + \dot{b} + \dot{c} + \dot{d} = n(n-1)/2$.

Proposition 5 *Given two PCs* U *and* V*, we have* $\dot{a} + \dot{b} + \dot{c} + \dot{d} + \dot{e} = \max\{\sum_{i<j} \mathrm{T}^{\mathrm{U}}_{i,j}, \sum_{i<j} \mathrm{T}^{\mathrm{V}}_{i,j}\}$.

Proof Let $\mathrm{M} \triangleq \max\{\mathrm{T}^{\mathrm{U}}, \mathrm{T}^{\mathrm{V}}\}$. If $\mathrm{T}^{\mathrm{U}}_{i,j} \geq \mathrm{T}^{\mathrm{V}}_{i,j}$, then $\min\{\mathrm{T}^{\mathrm{U}}_{i,j}, \mathrm{T}^{\mathrm{V}}_{i,j}\} + \mathrm{M}_{i,j} - \mathrm{T}^{\mathrm{V}}_{i,j} = \mathrm{T}^{\mathrm{U}}_{i,j}$. If $\mathrm{T}^{\mathrm{U}}_{i,j} < \mathrm{T}^{\mathrm{V}}_{i,j}$, then $\min\{\mathrm{T}^{\mathrm{U}}_{i,j}, \mathrm{T}^{\mathrm{V}}_{i,j}\} + \mathrm{M}_{i,j} - \mathrm{T}^{\mathrm{V}}_{i,j} = \mathrm{T}^{\mathrm{U}}_{i,j}$ as well. Thus, $\mathrm{T}^{\mathrm{U}}_{i,j} = \min\{\mathrm{T}^{\mathrm{U}}_{i,j}, \mathrm{T}^{\mathrm{V}}_{i,j}\} + \mathrm{M}_{i,j} - \mathrm{T}^{\mathrm{V}}_{i,j}$, and the same reasoning works for $\mathrm{T}^{\mathrm{V}}_{i,j} = \min\{\mathrm{T}^{\mathrm{U}}_{i,j}, \mathrm{T}^{\mathrm{V}}_{i,j}\} + \mathrm{M}_{i,j} - \mathrm{T}^{\mathrm{U}}_{i,j}$. We know that $\dot{a} + \dot{b} + \dot{c} + \dot{d} = \sum_{i<j} \min\{\mathrm{T}^{\mathrm{U}}_{i,j}, \mathrm{T}^{\mathrm{V}}_{i,j}\}$ by Proposition 4. If $\sum_{i<j} \mathrm{T}^{\mathrm{U}}_{i,j} \geq \sum_{i<j} \mathrm{T}^{\mathrm{V}}_{i,j}$, we have $\dot{e} = \sum_{i<j} (\mathrm{M}_{i,j} - \mathrm{T}^{\mathrm{V}}_{i,j})$ and $\dot{a} + \dot{b} + \dot{c} + \dot{d} + \dot{e} = \sum_{i<j} \mathrm{T}^{\mathrm{U}}_{i,j}$; otherwise, $\dot{a} + \dot{b} + \dot{c} + \dot{d} + \dot{e} = \sum_{i<j} \mathrm{T}^{\mathrm{V}}_{i,j}$. ■

Corollary 2 *The sum* $\dot{a} + \dot{b} + \dot{c} + \dot{d} + \dot{e}$ *is constant over all simultaneous permutations of the columns of* U *and* V *because they do not alter the sums* $\sum_{i<j} \mathrm{T}^{\mathrm{U}}_{i,j}$ *and* $\sum_{i<j} \mathrm{T}^{\mathrm{V}}_{i,j}$.

Corollary 3 *13FRI (13AFRI) and 13GRI (13AGRI) are equivalent when applied to FCs because* $\max\{\sum_{i<j} \mathrm{T}^{\mathrm{U}}_{i,j}, \sum_{i<j} \mathrm{T}^{\mathrm{V}}_{i,j}\} = n(n-1)/2 = \dot{a} + \dot{b} + \dot{c} + \dot{d}$.

Corollary 4 *Given two EHCs* U *and* V*, we have* $\mathrm{13GRI}(\mathrm{U}, \mathrm{V}) = \mathrm{RI}(\mathrm{U}, \mathrm{V})$ *because of Proposition 2 and Corollary 3.*

Corollary 5 *Given two EHCs* U *and* V*, we have* $\mathrm{13AGRI}(\mathrm{U}, \mathrm{V}) = \mathrm{ARI}(\mathrm{U}, \mathrm{V})$ *because of Proposition 3 and Corollary 3.*

References

Ahmed N. Albatineh, Magdalena Niewiadomska-Bugaj, and Daniel Mihalko. On similarity indices and correction for chance agreement. *Journal of Classification*, 23:301–313, 2006. 10.1007/s00357-006-0017-z.

Robert Alcock. Synthetic control chart time series data set, 1999. URL `http://archive.ics.uci.edu/ml/datasets/Synthetic+Control+Chart+Time+Series`.

Enrique Amigó, Julio Gonzalo, Javier Artiles, and Felisa Verdejo. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Inf. Retr.*, 12(4):461–486, August 2009. ISSN 1386-4564.

Derek T. Anderson, James C. Bezdek, Mihail Popescu, and James M. Keller. Comparing fuzzy, probabilistic, and possibilistic partitions. *IEEE Trans. Fuzzy Syst.*, 18(5):906–918, June 2010.

Derek T. Anderson, James M. Keller, Ozy Sjahputera, James C. Bezdek, and Mihail Popescu. Comparing soft clusters and partitions. In *Fuzzy Systems (FUZZ), 2011 IEEE International Conference on*, pages 924 –931, june 2011.

Amit Bagga and Breck Baldwin. Entity-based cross-document coreferencing using the vector space model. In *Proceedings of the 17th International Conference on Computational Linguistics - Volume 1*, COLING '98, pages 79–85, Stroudsburg, PA, USA, 1998. Association for Computational Linguistics.

Jrgen Beringer and Eyke Hllermeier. *Fuzzy Clustering of Parallel Data Streams*, pages 333–352. John Wiley & Sons, Ltd, 2007.

James C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Kluwer Academic Publishers, Norwell, MA, USA, 1981. ISBN 0306406713.

Christian Borgelt. Resampling for fuzzy clustering. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, pages 595–614, 2007.

Christian Borgelt and Rudolf Kruse. Finding the number of fuzzy clusters by resampling. In *Fuzzy Systems, 2006 IEEE International Conference on*, pages 48–54, 0-0 2006.

Roelof Brouwer. Extending the rand, adjusted rand and jaccard indices to fuzzy partitions. *Journal of Intelligent Information Systems*, 32:213–235, 2009. 10.1007/s10844-008-0054-7.

Ricardo J. G. B. Campello. A fuzzy extension of the rand index and other related indexes for clustering and classification assessment. *Pattern Recognition Letters*, 28(7):833 – 841, 2007.

Ricardo J. G. B. Campello. Generalized external indexes for comparing data partitions with overlapping categories. *Pattern Recognition Letters*, 31(9):966–975, 2010. ISSN 0167-8655.

Claire Cardie and Kiri Wagstaf. Noun phrase coreference as clustering. In *Proceedings of the 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC1999)*, pages 82–89, College Park, Maryland, USA, June 21–22 1999.

Michele Ceccarelli and Antonio Maratea. A fuzzy extension of some classical concordance measures and an efficient algorithm for their computation. In Ignac Lovrek, Robert

Howlett, and Lakhmi Jain, editors, *Knowledge-Based Intelligent Information and Engineering Systems*, volume 5179 of *Lecture Notes in Computer Science*, pages 755–763. Springer Berlin / Heidelberg, 2008.

Michele Ceccarelli and Antonio Maratea. Concordance indices for comparing fuzzy, possibilistic, rough and grey partitions. *Int. J. Knowl. Eng. Soft Data Paradigm.*, 1(4): 331–344, October 2009.

Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977. ISSN 00359246.

Brian S. Everitt, Sabine Landau, and Morven Leese. *Cluster Analysis*. Arnold Publishers, May 2001.

E. B. Fowlkes and C. L. Mallows. A Method for Comparing Two Hierarchical Clusterings. *Journal of the American Statistical Association*, 78(383):553–569, 1983.

Ana L. N. Fred and Anil K. Jain. Combining multiple clusterings using evidence accumulation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(6):835–850, 2005. ISSN 0162-8828.

Stephan Günnemann, Ines Färber, Emmanuel Müller, Ira Assent, and Thomas Seidl. External evaluation measures for subspace clustering. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11, pages 1363–1372, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0717-8.

Trevor Hastie, Robert Tibshirani, Gavin Sherlock, Michael Eisen, Patrick Brown, and David Botstein. Imputing missing data for gene expression arrays. Technical report, Stanford University, 1999.

Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2 (1):193–218, December 1985.

Eyke Hullermeier and Maria Rifqi. A fuzzy variant of the rand index for comparing clustering structures. In *Proc. IFSA*, page 16, Lisbon, Portugal, 2009.

Paul Jaccard. Nouvelles recherches sur la distribution florale. *Bulletin de la Socit Vaudoise de Sciences Naturelles*, 44:223–370, 1908.

Anil K. Jain and Richard C. Dubes. *Algorithms for Clustering Data*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988. ISBN 0-13-022278-X.

Ian T. Jolliffe. *Principal Component Analysis*. Springer, second edition, October 2002.

Karin Kailing, Hans-Peter Kriegel, and Peer Krger. Density-connected subspace clustering for high-dimensional data. In *Proceedings SDM (2004)*, pages 246–257, 2004.

Leonard Kaufman and Peter J. Rousseeuw. *Finding Groups in Data: an Introduction to Cluster Analysis*. Wiley, 1990.

Ludmila I. Kuncheva and Dmitry P. Vetrov. Evaluation of stability of k-means cluster ensembles with respect to random initialization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(11):1798 –1808, nov. 2006.

Vladimir I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8):707–710, 1966.

Ping Luo, Hui Xiong, Guoxing Zhan, Junjie Wu, and Zhongzhi Shi. Information-theoretic distance measures for clustering validation: Generalization and normalization. *IEEE Trans. on Knowl. and Data Eng.*, 21(9):1249–1262, September 2009.

David J. C. Mackay. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, 1 edition, June 2003. ISBN 0521642981.

James B. MacQueen. Some methods for classification and analysis of multivariate observations. In L. M. Le Cam and J. Neyman, editors, *Proc. of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967.

Sara C. Madeira and Arlindo L. Oliveira. Biclustering algorithms for biological data analysis: A survey. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, 1(1):24–45, 2004.

Marina Meila. Comparing clusterings by the variation of information. In Bernhard Schlkopf and Manfred Warmuth, editors, *Learning Theory and Kernel Machines*, volume 2777 of *Lecture Notes in Computer Science*, pages 173–187. Springer Berlin / Heidelberg, 2003. ISBN 978-3-540-40720-1.

Marina Meila. Comparing clusterings: an axiomatic view. In *Proceedings of the 22nd International Conference on Machine Learning*, ICML '05, pages 577–584, New York, NY, USA, 2005. ACM.

Marina Meila. Comparing clusterings—an information based distance. *Journal of Multivariate Analysis*, 98:873–895, May 2007.

Marina Meila. Local equivalences of distances between clusterings–a geometric perspective. *Machine Learning*, 86(3):369–389, March 2012.

Stefano Monti, Pablo Tamayo, Jill Mesirov, and Todd Golub. Consensus clustering: A resampling-based method for class discovery and visualization of gene expression microarray data. *Machine Learning*, 52:91–118, July 2003.

David Newman and Arthur Asuncion. UCI machine learning repository, 2010. URL `http://archive.ics.uci.edu/ml`.

Malay K. Pakhira, Sanghamitra Bandyopadhyay, and Ujjwal Maulik. A study of some fuzzy cluster validity indices, genetic clustering and application to pixel classification. *Fuzzy Sets and Systems*, 155(2):191 – 214, 2005. ISSN 0165-0114.

Nikhil R. Pal and James C. Bezdek. On cluster validity for the fuzzy c-means model. *IEEE Trans. on Fuzzy Systems*, 3(3):370–379, 1995.

Anne Patrikainen and Marina Meila. Comparing subspace clusterings. *IEEE Trans. on Knowl. and Data Eng.*, 18(7):902–916, 2006.

Romain Quere and Carl Frelicot. A normalized soft window-based similarity measure to extend the rand index. In *Fuzzy Systems (FUZZ), 2011 IEEE International Conference on*, pages 2513 –2520, june 2011.

Romain Quere, Hoel Le Capitaine, Noel Fraisseix, and Carl Frelicot. On normalizing fuzzy coincidence matrices to compare fuzzy and/or possibilistic partitions with the rand index. In *Proceedings of the 2010 IEEE International Conference on Data Mining*, pages 977–982, Washington, DC, USA, 2010. IEEE Computer Society.

William M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850, 1971. ISSN 01621459.

Stefano Rovetta and Francesco Masulli. An experimental validation of some indexes of fuzzy clustering similarity. In *Proceedings of the 8th International Workshop on Fuzzy Logic and Applications*, WILF '09, pages 132–139, Berlin, Heidelberg, 2009. Springer-Verlag.

Douglas Steinley. Properties of the hubert-arabie adjusted rand index. *Psychological Methods*, 9(3):386–396, 2004.

Alexander Strehl and Joydeep Ghosh. Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3:583–617, March 2003.

Nguyen X. Vinh and Julien Epps. A novel approach for automatic number of clusters detection in microarray data based on consensus clustering. In *Proceedings of the 2009 Ninth IEEE International Conference on Bioinformatics and Bioengineering*, BIBE '09, pages 84–91, Washington, DC, USA, 2009. IEEE Computer Society.

Nguyen X. Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: is a correction for chance necessary? In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 1073–1080, New York, NY, USA, 2009. ACM.

Nguyen X. Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research*, 11:2837–2854, 2010.

Zhimin Wang. Metrics for overlapping clustering comparison, November 2010. URL `http://etaxonomy.org/mw/File:Sigs.pdf`.

Zhimin Wang. Entropy on covers. *Data Mining and Knowledge Discovery*, 24:288–309, 2012. ISSN 1384-5810. 10.1007/s10618-011-0230-1.

Junjie Wu, Hui Xiong, and Jian Chen. Adapting the right measures for k-means clustering. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 877–886, New York, NY, USA, 2009. ACM.

Kuo-Lung Wu and Miin-Shen Yang. A cluster validity index for fuzzy clustering. *Pattern Recognition Letters*, 26:1275–1291, July 2005. ISSN 0167-8655.

Jian Yu, Qiansheng Cheng, and Houkuan Huang. Analysis of the weighting exponent in the fcm. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 34(1):634 – 639, feb. 2004.

Zhiwen Yu, Hau-San Wong, and Hongqiang Wang. Graph-based consensus clustering for class discovery from gene expression data. *Bioinformatics*, 23(21):2888–2896, 2007.

Jiang-She Zhang and Yiu-Wing Leung. Improved possibilistic c-means clustering algorithms. *Fuzzy Systems, IEEE Transactions on*, 12(2):209–217, april 2004.

Shaohong Zhang, Hau-San Wong, and Ying Shen. Generalized adjusted rand indices for cluster ensembles. *Pattern Recognition*, 45(6):2214 – 2226, 2012.

Journal of Machine Learning Research 16 (2015) 2999-3034 Submitted 6/14; Revised 4/15; Published 7/15

Completing Any Low-rank Matrix, Provably*

Yudong Chen YUDONG.CHEN@EECS.BERKELEY.EDU
Department of Electrical Engineering and Computer Sciences
University of California, Berkeley
Berkeley, CA 94704, USA

Srinadh Bhojanapalli BSRINADH@UTEXAS.EDU
Sujay Sanghavi SANGHAVI@MAIL.UTEXAS.EDU
Department of Electrical and Computer Engineering
The University of Texas at Austin
Austin, TX 78712, USA

Rachel Ward RWARD@MATH.UTEXAS.EDU
Department of Mathematics and ICES
The University of Texas at Austin
Austin, TX 78712, USA

Editor: Tong Zhang

Abstract

Matrix completion, i.e., the exact and provable recovery of a low-rank matrix from a small subset of its elements, is currently only known to be possible if the matrix satisfies a restrictive structural constraint—known as *incoherence*—on its row and column spaces. In these cases, the subset of elements is assumed to be sampled uniformly at random.

In this paper, we show that *any* rank-r n-by-n matrix can be exactly recovered from as few as $O(nr\log^2 n)$ randomly chosen elements, provided this random choice is made according to a *specific biased distribution* suitably dependent on the coherence structure of the matrix: the probability of any element being sampled should be at least a constant times the sum of the leverage scores of the corresponding row and column. Moreover, we prove that this specific form of sampling is nearly necessary, in a natural precise sense; this implies that many other perhaps more intuitive sampling schemes fail.

We further establish three ways to use the above result for the setting when leverage scores are not known *a priori*. (a) We describe a provably-correct sampling strategy for the case when only the column space is incoherent and no assumption or knowledge of the row space is required. (b) We propose a two-phase sampling procedure for general matrices that first samples to estimate leverage scores followed by sampling for exact recovery. These two approaches assume control over the sampling procedure. (c) By using our main theorem in a reverse direction, we provide an analysis showing the advantages of the (empirically successful) weighted nuclear/trace-norm minimization approach over the vanilla un-weighted formulation given non-uniformly distributed observed elements. This approach does not require controlled sampling or knowledge of the leverage scores.

Keywords: matrix completion, coherence, leverage score, nuclear norm, weighted nuclear norm

∗. Partial preliminary results appeared at the International Conference on Machine Learning (ICML) 2014 under the title "Coherent Matrix Completion".

1. Introduction

Low-rank matrix completion has been the subject of much recent study due to its application in myriad tasks: collaborative filtering, dimensionality reduction, clustering, non negative matrix factorization and localization in sensor networks. Clearly, the problem is ill-posed in general; correspondingly, analytical work on the subject has focused on the joint development of algorithms, and sufficient conditions under which such algorithms are able to recover the matrix.

While they differ in scaling/constant factors, all existing sufficient conditions (Candès and Recht, 2009; Candès and Tao, 2010; Recht, 2011; Keshavan et al., 2010; Gross, 2011; Jain et al., 2013; Negahban and Wainwright, 2012)—with a couple of exceptions we describe in Section 2—require that *(a)* the subset of observed elements should be uniformly randomly chosen, independent of the values of the matrix elements, and *(b)* the low-rank matrix be "incoherent" or "not spiky"—i.e., its row and column spaces should be diffuse, having low inner products with the standard basis vectors. Under these conditions, the matrix has been shown to be provably recoverable—via methods based on convex optimization (Candès and Recht, 2009), alternating minimization (Jain et al., 2013), iterative thresholding (Cai et al., 2010), etc.—using as few as $\Theta(nr \log n)$ observed elements for an $n \times n$ matrix of rank r.

Actually, the incoherence assumption *is required because of the uniform sampling:* coherent matrices are those which have most of their mass in a relatively small number of elements. By sampling entries uniformly and independently at random, most of the mass of a coherent low-rank matrix will be missed; this could (and *does*) throw off most existing methods for exact matrix completion. One could imagine that if the sampling is adapted to the matrix, roughly in a way that ensures that elements with more mass are more likely to be observed, then it may be possible for *existing* methods to recover the full matrix.

In this paper, we show that the incoherence requirement can be eliminated completely, provided the sampling distribution is dependent on the matrix to be recovered in the right way. Specifically, we have the following results.

1. If the probability of an element being observed is proportional to the sum of the corresponding row and column leverage scores (which are local versions of the standard incoherence parameter) of the underlying matrix, then an *arbitrary* rank-r matrix can be exactly recovered from $\Theta(nr \log^2 n)$ observed elements with high probability, using nuclear norm minimization (Theorem 2 and Corollary 3). In the case when all leverage scores are uniformly bounded from above, our results reduce to existing guarantees for incoherent matrices using uniform sampling. Our sample complexity bound $\Theta(nr \log^2 n)$ is optimal up to a single factor of $\log^2 n$, since the degrees of freedom in an $n \times n$ matrix of rank r is in general in the order of nr. Moreover, we show that to complete a coherent matrix, it is *necessary* (in certain precise sense) to sample according to the leverage scores as above (Theorem 6).

2. For a matrix whose column space is incoherent and row space is arbitrarily coherent, our results immediately lead to a provably correct sampling scheme which *requires no prior knowledge of the leverage scores of the underlying matrix* and has near optimal sample complexity (Corollary 4).

3. We provide numerical evidence that a two-phase adaptive sampling strategy, which assumes no prior knowledge about the leverage scores of the underlying matrix, can perform on par with the optimal sampling strategy in completing coherent matrices, and significantly outperforms uniform sampling (Section 4). Specifically, we consider a two-phase sampling strategy whereby given a fixed budget of m samples, we first draw a fixed proportion of samples uniformly at random, and then draw the remaining samples according to the leverage scores of the resulting sampled matrix.

4. As a corollary of our main theorem, we are able to obtain the first exact recovery guarantee for the *weighted* nuclear norm minimization approach, which can be viewed as adjusting the leverage scores to align with the given sampling distribution. Our results provide a strategy for choosing the weights when non-uniformly distributed samples are *given* so as to order-wise reduce the sample complexity of the weighted approach to that of the standard *unweighted* formulation (Theorem 7). Our theorem quantifies the benefit of the weighted approach, thus providing theoretical justification for its good empirical performance observed in Srebro and Salakhutdinov (2010); Foygel et al. (2011); Negahban and Wainwright (2012).

These results provide a deeper and more general theoretical understanding of the relation between the sampling procedure and the matrix coherence/leverage-score structure, and how they affect the recovery performance. While in practice one may not have complete control over the sampling procedure, or exact knowledge of the matrix leverage scores, partial control and knowledge are often possible, and we believe our theory provides useful approximations and insights. We expect that the ideas and results in this paper will serve as the foundation for developing algorithms for more general settings and applications.

Our theoretical results are achieved by a new analysis based on concentration bounds involving the *weighted* $\ell_{\infty,2}$ matrix norm, defined as the maximum of the appropriately weighted row and column norms of the matrix. This differs from previous approaches that use ℓ_∞ or unweighted $\ell_{\infty,2}$ norm bounds (Gross, 2011; Recht, 2011; Chen, 2015). In some sense, using the weighted $\ell_{\infty,2}$-type bounds is natural for the analysis of low-rank matrix recover/approximation when the observations are in the form of entries of rows/columns of the matrix, because the rank is a property of the rows and columns of the matrix rather than its individual elements, and the weighted norm captures the relative importance of the rows/columns. Therefore, our techniques based on the $\ell_{\infty,2}$ norm might be of independent interest beyond the specific settings and algorithms considered here.

1.1 Organization

In Section 2 we briefly survey the relevant literature. We present our main results for coherent matrix completion in Section 3. In Section 4 we propose a two-phase algorithm that requires no prior knowledge about the underlying matrix's leverage scores. In Section 5 we provide guarantees for weighted nuclear norm minimization. The paper concludes with a discussion of future work in Section 6. We provide the proofs of the main theorems in the appendix.

2. Related Work

There is now a vast body of literature on matrix completion, and an even bigger body of literature on matrix approximations; we restrict our literature review here to papers that are most directly related.

Completion of incoherent and row-coherent matrices: The first algorithm and theoretical guarantees for exact low-rank matrix completion appeared in Candès and Recht (2009); there it was shown that nuclear norm minimization works when the low-rank matrix is incoherent, and the sampling is uniform random and independent of the matrix. Subsequent works have refined provable completion results for incoherent matrices under the uniform random sampling model, both via nuclear norm minimization (Candès and Tao, 2010; Recht, 2011; Gross, 2011; Chen, 2015), and other methods like SVD followed by local descent (Keshavan et al., 2010) and alternating minimization (Jain et al., 2013), etc. The setting with sparse errors and additive noise is also considered (Candès and Plan, 2010; Chandrasekaran et al., 2011; Chen et al., 2013; Candès et al., 2011; Negahban and Wainwright, 2012).

The recent work in Krishnamurthy and Singh (2013) considers matrix completion when the row space is allowed to be coherent but the column space is still required to be incoherent with parameter μ_0. Their proposed adaptive sampling algorithm selects columns to observe in their entirety and requires a total of $O(\mu_0 r^{3/2} n \log(2r/\delta))$ observed elements with a success probability $1-\delta$, which is superlinear in r. A corollary of our results guarantees a sample complexity that is linear in r in this row-coherent setting. The sample complexity was recently improved to $O(\mu_0 rn \log^2(r^2/\delta))$ in Krishnamurthy and Singh (2014).

Matrix approximations via sub-sampling: Weighted sampling methods have been widely considered in the related context of matrix *sparsification*, where one aims to approximate a given large dense matrix with a sparse matrix. The strategy of element-wise matrix sparsification was introduced in Achlioptas and McSherry (2007). They propose and provide bounds for the ℓ_2 *element-wise sampling* model, where elements of the matrix are sampled with probability proportional to their squared magnitude. These bounds were later refined in Drineas and Zouzias (2011). Alternatively, Arora et al. (2006) propose the ℓ_1 *element-wise sampling* model, where elements are sampled with probabilities proportional to their magnitude. This model was further investigated in Achlioptas et al. (2013) and argued to be almost always preferable to ℓ_2 sampling.

Closely related to the matrix sparsification problem is the matrix *column selection* problem, where one aims to find the "best" k column subset of a matrix to use as an approximation. State-of-the-art algorithms for column subset selection (Boutsidis et al., 2009; Mahoney, 2011) involve randomized sampling strategies whereby columns are selected proportionally to their *statistical leverage scores*—the squared Euclidean norms of projections of the canonical unit vectors on the column subspaces. The statistical leverage scores of a matrix can be approximated efficiently, faster than the time needed to compute an SVD (Drineas et al., 2012). Statistical leverage scores are also used extensively in statistical regression analysis for outlier detection (Chatterjee and Hadi, 1986). More recently, statistical leverage scores were used in the context of graph sparsification under the name of graph resistance (Spielman and Srivastava, 2011). The sampling distribution we use for the

matrix completion guarantees of this paper is *elemen-wise* and based on statistical leverage scores. As shown both theoretically (Theorem 6) and empirically (Section 4.1), sampling as such outperforms both ℓ_1 and ℓ_2 element-wise sampling, at least in the context of matrix completion.

Weighted sampling in compressed sensing: This paper is similar in spirit to recent work in compressed sensing which shows that sparse recovery guarantees traditionally requiring mutual incoherence can be extended to systems which are only *weakly* incoherent, without any loss of approximation power, provided measurements from the sensing basis are subsampled according to their coherence with the sparsity basis. This notion of *local coherence sampling* seems to have originated in Rauhut and Ward (2012) in the context of sparse orthogonal polynomial expansions, and has found applications in uncertainty quantification (Yang and Karniadakis, 2013), interpolation with spherical harmonics (Burq et al., 2012), and MRI compressive imaging (Krahmer and Ward, 2014).

3. Main Results

The results in this paper hold for what is arguably the most popular approach to matrix completion: nuclear norm minimization. If the true matrix is M with its (i,j)-th element denoted by M_{ij}, and the set of observed elements is Ω, this method estimates M via the optimum of the convex program:

$$\begin{aligned} \min_X \quad & \|X\|_* \\ \text{s.t.} \quad & X_{ij} = M_{ij} \text{ for } (i,j) \in \Omega. \end{aligned} \tag{1}$$

where the nuclear norm $\|\cdot\|_*$ of a matrix is the sum of its singular values.[1]

We focus on the setting where matrix elements are revealed according an underlying probability distribution. To introduce the distribution of interest, we first need a definition.

Definition 1 (Leverage Scores) *For an $n_1 \times n_2$ real-valued matrix M of rank r whose rank-r SVD is given by $U\Sigma V^\top$, its (normalized) leverage scores—$\mu_i(M)$ for any row i, and $\nu_j(M)$ for any column j—are defined as*

$$\begin{aligned} \mu_i(M) &:= \frac{n_1}{r}\left\|U^\top e_i\right\|_2^2, \quad i = 1, 2, \ldots, n_1, \\ \nu_j(M) &:= \frac{n_2}{r}\left\|V^\top e_j\right\|_2^2, \quad j = 1, 2, \ldots, n_2, \end{aligned} \tag{2}$$

where e_i denotes the i-th standard basis element with appropriate dimension.[2]

Note that the leverage scores are non-negative, and are functions of the column and row spaces of the matrix M. Since U and V have orthonormal columns, we always have relationship $\sum_i \mu_i(M) r/n_1 = \sum_j \nu_j(M) r/n_2 = r$. The standard *incoherence parameter* μ_0

1. This becomes the trace norm for positive-definite matrices. It is now well-recognized to be a convex surrogate for the rank function (Fazel, 2002).
2. In the matrix sparsification literature (Drineas et al., 2012; Boutsidis et al., 2009) and beyond, the leverage scores of M often refer to the *un-normalized* quantities $\left\|U^\top e_i\right\|^2$ and $\left\|V^\top e_j\right\|^2$.

of M used in the previous literature corresponds to a global upper bound on the leverage scores:

$$\mu_0 \geq \max_{i,j}\{\mu_i(M), \nu_j(M)\}.$$

Therefore, the leverage scores can be considered as the localized versions of the standard incoherence parameter.

We are ready to state our main result, the theorem below.

Theorem 2 *Let $M = (M_{ij})$ be an $n_1 \times n_2$ matrix of rank r, and suppose that its elements M_{ij} are observed only over a subset of elements $\Omega \subset [n_1]\times[n_2]$. There is a universal constant $c_0 > 0$ such that, if each element (i, j) is independently observed with probability p_{ij}, and p_{ij} satisfies*

$$\begin{aligned} p_{ij} &\geq \min\left\{ c_0 \frac{(\mu_i(M) + \nu_j(M))\, r \log^2(n_1+n_2)}{\min\{n_1, n_2\}},\quad 1 \right\}, \qquad (3)\\ p_{ij} &\geq \frac{1}{\min\{n_1, n_2\}^{10}}, \end{aligned}$$

then M is the unique optimal solution to the nuclear norm minimization problem (1) with probability at least $1 - 5(n_1 + n_2)^{-10}$.

We will refer to the sampling strategy (3) as *leveraged sampling*. Note that the expected number of observed elements is $\sum_{i,j} p_{ij}$, and this satisfies

$$\begin{aligned} \sum_{i,j} p_{ij} &\geq \max\left\{ c_0 \frac{r\log^2(n_1+n_2)}{\min\{n_1,n_2\}} \sum_{i,j} \left(\mu_i(M) + \nu_j(M)\right), \sum_{i,j} \frac{1}{\min\{n_1,n_2\}^{10}} \right\} \\ &= 2c_0 \max\{n_1, n_2\}\, r \log^2(n_1+n_2), \end{aligned}$$

which is independent of the leverage scores, or indeed any other property of the matrix. Hoeffding's inequality implies that the actual number of observed elements sharply concentrates around its expectation, leading to the following corollary:

Corollary 3 *Let $M = (M_{ij})$ be an $n_1 \times n_2$ matrix of rank r. Draw a subset Ω of its elements by leveraged sampling according to the procedure described in Theorem 2. There is a universal constant $c_0 > 0$ such that the following holds with probability at least $1 - 10(n_1 + n_2)^{-10}$: the number m of revealed elements is bounded by*

$$|\Omega| \leq 3c_0 \max\{n_1, n_2\}\, r \log^2(n_1+n_2)$$

and M is the unique optimal solution to the nuclear norm minimization program (1).

We now provide comments and discussion.

(A) Roughly speaking, the condition given in (3) ensures that elements in important rows/columns (indicated by large leverage scores μ_i and ν_j) of the matrix should be observed more often. Note that Theorem 2 only stipulates that an *inequality* relation hold between p_{ij} and $\{\mu_i(M), \nu_j(M)\}$. This allows for there to be some discrepancy between the sampling

distribution and the leverage scores. It also has the natural interpretation that the more the sampling distribution $\{p_{ij}\}$ is "aligned" to the leverage score pattern of the matrix, the fewer observations are needed.

(B) Sampling based on leverage scores provides close to the optimal number of sampled elements required for exact recovery (when sampled with any distribution). In particular, recall that the number of degrees of freedom of an $n \times n$ matrix of rank r is $2nr(1 - r/2n)$, and knowing the leverage scores of the matrix reduces the degrees of freedom by $2n$ in the worst case. Hence, regardless of how the elements are sampled, a minimum of $\Theta(nr)$ elements is required to recover the matrix. Theorem 2 matches this lower bound, with an additional $O(\log^2(n))$ factor.

(C) Our work improves on existing results *even* in the case of uniform sampling and uniform incoherence. Recall that the original work of Candès and Recht (2009), and subsequent works (Candès and Tao, 2010; Recht, 2011; Gross, 2011) give recovery guarantees based on two parameters of the matrix $M \in \mathbb{R}^{n\times n}$ (assuming its SVD is $U\Sigma V^\top$): (a) the (above-defined) *incoherence parameter* μ_0, which is a uniform bound on the leverage scores, and (b) a *joint incoherence parameter* μ_{str} defined by $\|UV^\top\|_\infty = \sqrt{\frac{r\mu_{\text{str}}}{n^2}}$. With these definitions, the current state of the art states that if the sampling probability is uniform and satisfies

$$p_{ij} \equiv p \geq c\frac{\max\{\mu_0, \mu_{\text{str}}\} r \log^2 n}{n}, \quad \forall i, j,$$

where c is a constant, then M will be the unique optimum of (1) with high probability. A direct corollary of our work improves on this result, by removing the need for extra constraints on the joint incoherence; in particular, it is easy to see that our theorem implies that a uniform sampling probability of $p \geq c\frac{\mu_0 r \log^2 n}{n}$—that is, with no μ_{str}—guarantees recovery of M with high probability. Note that μ_{str} can be as high as $\mu_0 r$, for example, in the case when M is positive semi-definite; our corollary thus removes this sub-optimal dependence on the rank and on the incoherence parameter. This improvement was recently observed in Chen (2015).

3.1 Knowledge-Free Completion for Row Coherent Matrices

Theorem 2 immediately yields a useful result in scenarios where only the row space of a matrix is coherent and one has control over the sampling of the matrix. This setting is considered by Krishnamurthy and Singh (2013).

Suppose the column space of $M \in \mathbb{R}^{n\times n}$ is incoherent with $\max_i \mu_i(M) \leq \mu_0$ and the row space is arbitrary (we consider square matrix for simplicity). For a number $0 < \delta < 1$ to be prescribed by the user, We choose each row of M with probability $\frac{10\mu_0 r}{n} \log \frac{2r}{\delta}$, and observe all the elements of the chosen rows. We then compute the leverage scores $\{\tilde{\nu}_j\}$ of the space spanned by these rows, and use them as estimates for $\nu_j(M)$, the leverage scores of M. Based on these estimates, we can perform leveraged sampling according to (3) and then use nuclear norm minimization to recover M. Note that this procedure does not require any prior knowledge about the leverage scores of M. The following corollary shows that the procedure is *provably correct* and exactly recovers M with high probability, using a near-optimal number of samples.

Corollary 4 *For any number $0 < \delta < 1$ and some universal constants $c_0, c_1 > 0$, the following holds. With probability at least $1 - \delta$, the above procedure computes the column leverage scores of M exactly, i.e., $\tilde{\nu}_j = \nu_j(M), \forall j \in [n]$. If we set $\delta = 4n^{-10}$, and further sample a set Ω of elements of M with probabilities*

$$p_{ij} = \min\left\{c_0 \frac{(\mu_0 + \tilde{\nu}_j) r \log^2 n}{n}, 1\right\}, \quad \forall i, j,$$

then with probability at least $1 - 10n^{-10}$, M is the unique optimal solution to the nuclear norm minimization program (1), and we use a total of at most $c_1 \mu_0 rn \log^2 n$ samples.

The algorithm proposed in Krishnamurthy and Singh (2013) requires a sample complexity of $O(\mu_0 r^{3/2} n \log(2r/\delta))$ (and guarantees a success probability of $1 - \delta$). Our result in the corollary above removes the sub-optimal $r^{3/2}$ factor in the sample complexity. Very recently Krishnamurthy and Singh (2014) provide a new sample complexity bound $O(\mu_0 rn \log^2(r^2/\delta))$ using the same algorithm from their previous paper. We note that our sampling strategy is different from theirs: we sample entire rows of M, whereas they sample entire columns.

3.2 Necessity of Leveraged Sampling

In this subsection, we show that the leveraged sampling in (3) is necessary for completing a coherent matrix in a certain precise sense. For simplicity, we restrict ourselves to square matrices in $\mathbb{R}^{n \times n}$. Suppose each element (i, j) is observed independently with probability p_{ij}. We consider a family of sampling probabilities $\{p_{ij}\}$ with the following property.

Definition 5 (Location Invariance) *$\{p_{ij}\}$ is said to be location-invariant with respect to the matrix M if the following are satisfied: (1) For any two rows $i \neq i'$ that are identical, i.e., $M_{ij} = M_{i'j}$ for all j, we have $p_{ij} = p_{i'j}$ for all j; (2) For any two columns $j \neq j'$ that are identical, i.e., $M_{ij} = M_{ij'}$ for all i, we have $p_{ij} = p_{ij'}$ for all i.*

In other words, $\{p_{ij}\}$ is location-invariant with respect to M if identical rows (or columns) of M have identical sampling probabilities. We consider this assumption very mild, and it covers the leveraged sampling as well as many other typical sampling schemes, including:

- uniform sampling, where $p_{ij} \equiv p$,
- element-wise magnitude sampling, where $p_{ij} \propto |M_{ij}|$ (ℓ_1 sampling) or $p_{ij} \propto M_{ij}^2$ (ℓ_2 sampling), and
- row/column-wise magnitude sampling, where $p_{ij} \propto f\left(\|M_{i\cdot}\|_2, \|M_{\cdot j}\|_2\right)$ for some (usually coordinate-wise non-decreasing) function $f : \mathbb{R}_+^2 \mapsto [0, 1]$.

Given two n-dimensional vectors $\vec{\mu} = (\mu_1, \ldots, \mu_n)$ and $\vec{\nu} = (\nu_1, \ldots, \nu_n)$, we use $\mathcal{M}_r(\vec{\mu}, \vec{\nu})$ to denote the set of rank-r matrices whose leverage scores are bounded by $\vec{\mu}$ and $\vec{\nu}$; that is,

$$\mathcal{M}_r(\vec{\mu}, \vec{\nu}) := \left\{M \in \mathbb{R}^{n \times n} : \mathrm{rank}(M) = r; \mu_i(M) \leq \mu_i, \nu_j(M) \leq \nu_j, \forall i, j\right\}.$$

We have the following results.

Theorem 6 *Suppose $n \geq r \geq 2$. Given any $2r$ numbers $a_1, \ldots, a_r$ and $b_1, \ldots, b_r$ with $\frac{r}{4} \leq \sum_{k=1}^{r} \frac{1}{a_k}, \sum_{k=1}^{r} \frac{1}{b_k} \leq r$ and $\frac{2}{r} \leq a_k, b_k \leq \frac{2n}{r}, \forall k \in [r]$, there exist two n-dimensional vectors $\vec{\mu}$ and $\vec{\nu}$ and the corresponding set $\mathcal{M}_r(\vec{\mu}, \vec{\nu})$ with the following properties:*

1. *For each $i, j \in [n]$, $\mu_i = a_k$ and $\nu_j = b_{k'}$ for some $k, k' \in [r]$. That is, the values of the leverage scores are given by $\{a_k\}$ and $\{b_{k'}\}$.*

2. *There exists a matrix $M^{(0)} \in \mathcal{M}_r(\vec{\mu}, \vec{\nu})$ for which the following holds. If $\{p_{ij}\}$ is location-invariant w.r.t. $M^{(0)}$, and for some (i_0, j_0),*

$$p_{i_0 j_0} \leq \frac{\mu_{i_0} + \nu_{j_0}}{4n} \cdot r \log\left(\frac{2n}{(\mu_{i_0} \vee \nu_{j_0}) r}\right), \text{[3]} \tag{4}$$

   then with probability at least $\frac{1}{4}$, the following conclusion holds: There are infinitely many matrices $M^{(1)} \neq M^{(0)}$ in $\mathcal{M}_r(\vec{\mu}, \vec{\nu})$ such that $\{p_{ij}\}$ is location-invariant w.r.t. $M^{(1)}$, and

$$M^{(0)}_{ij} = M^{(1)}_{ij}, \quad \forall (i, j) \in \Omega.$$

3. *If we replace the condition (4) with*

$$p_{i_0 j_0} \leq \frac{\mu_{i_0} + \nu_{j_0}}{4n} \cdot r \log\left(\frac{n}{2}\right), \tag{5}$$

   then the conclusion above holds with probability at least $\frac{1}{n}$.

In other words, if (4) holds, then with probability at least $1/4$, no method can distinguish between $M^{(0)}$ and $M^{(1)}$; similarly, if (5) holds, then with probability at least $1/n$ no method succeeds. We shall compare these results with Theorem 2, which guarantees that if we use leveraged sampling,

$$p_{ij} \geq c_0 \frac{\mu_i + \nu_j}{n} \cdot r \log n, \quad \forall i, j$$

for some universal constant c_0, then for any matrix $M^{(0)}$ in $\mathcal{M}_r(\vec{\mu}, \vec{\nu})$, the nuclear norm minimization approach (1) recovers $M^{(0)}$ from its observed elements with failure probability no more than $\frac{1}{n}$. Therefore, under the setting of Theorem 6, leveraged sampling is *sufficient and necessary* for matrix completion up to one logarithmic factor for a target failure probability $\frac{1}{n}$ (or up to two logarithmic factors for a target failure probability $\frac{1}{4}$).

Admittedly, the setting covered by Theorem 6 has several restrictions on the sampling distributions and the values of the leverage scores. Nevertheless, we believe this result captures some essential difficulties in recovering general coherent matrices, and highlights how the sampling probabilities should relate in a specific way to the leverage score structure of the underlying object.

4. A Two-Phase Sampling Procedure

We have seen that one can exactly recover an arbitrary $n \times n$ rank-r matrix using $\Theta(nr \log^2 n)$ elements if sampled in accordance with the leverage scores. In practical applications of

3. We use the notation $a \vee b = \max\{a, b\}$.

matrix completion, even when the user is free to choose how to sample the matrix elements, she may not be privy to the leverage scores $\{\mu_i(M), \nu_j(M)\}$. In this section we propose a two-phase sampling procedure, described below and in Algorithm 1, which assumes no a priori knowledge about the matrix leverage scores, yet is observed to be competitive with the "oracle" leveraged sampling distribution (3).

Suppose we are given a total budget of m samples. The first step of the algorithm is to use the first β fraction of the budget to estimate the leverage scores of the underlying matrix, where $\beta \in [0, 1]$. Specifically, take a set of indices Ω sampled uniformly without replacement[4] such that $|\Omega| = \beta m$, and let $\mathcal{P}_\Omega(\cdot)$ be the sampling operator which maps the matrix elements not in Ω to 0. Take the rank-r SVD of $\mathcal{P}_\Omega(M)$, $\tilde{U}\tilde{\Sigma}\tilde{V}^\top$, where $\tilde{U}, \tilde{V} \in \mathbb{R}^{n\times r}$ and $\tilde{\Sigma} \in \mathbb{R}^{r\times r}$, and then use the leverage scores $\tilde{\mu}_i := \mu_i(\tilde{U}\tilde{\Sigma}\tilde{V}^\top)$ and $\tilde{\nu}_j := \nu_j(\tilde{U}\tilde{\Sigma}\tilde{V}^\top)$ as estimates for the column and row leverage scores of M. Now as the second step, generate the remaining $(1-\beta)m$ samples of the matrix M by sampling without replacement with distribution

$$\tilde{p}_{ij} \propto \frac{(\tilde{\mu}_i + \tilde{\nu}_j) r \log^2(2n)}{n}. \tag{6}$$

Let $\tilde{\Omega}$ denote the new set of samples. Using the combined set of samples $\mathcal{P}_{\Omega\cup\tilde{\Omega}}(M)$ as constraints, run the nuclear norm minimization program (1). Let $\hat{M}$ be the optimum of this program.

This approach of adjusting the sampling distribution based on leverage scores is relevant whenever we have some freedom in choosing the observed entries. For example, many recommendation systems do actively solicit users' opinions on some items chosen by the system, e.g., by asking them to fill out a survey or to choose from a list of items. While our assumptions are not strictly satisfied in practice, they are useful approximations and provide guidance for designing/analyzing practical systems. For example, in many systems there exist popular items that are viewed/rated by a large number of users, and "heavy" users that view/rate a large number of items. Our row-wise sampling procedure discussed in Section 3.1 can be viewed as an approximation of such settings.

To understand the performance of the two-phase algorithm, assume that the initial set of $m_1 = \beta m$ samples $\mathcal{P}_\Omega(M)$ are generated uniformly at random. If the underlying matrix

4. Note that sampling without replacement has lesser failure probability than the equivalent binomial sampling with replacement (Recht, 2011).

Algorithm 1 Two-phase sampling for coherent matrix completion

input Rank parameter r, sample budget m, and parameter $\beta \in [0, 1]$

Step 1: Obtain the initial set Ω by sampling uniformly without replacement such that $|\Omega| = \beta m$. Compute best rank-r approximation to $\mathcal{P}_\Omega(M)$, $\tilde{U}\tilde{\Sigma}\tilde{V}^\top$, and its leverage scores $\{\tilde{\mu}_i\}$ and $\{\tilde{\nu}_j\}$.

Step 2: Generate set of $(1-\beta)m$ new samples $\tilde{\Omega}$ by sampling without replacement with distribution (6). Set

$$\hat{M} = \arg\min_X \|X\|_* \text{ s.t } \mathcal{P}_{\Omega\cup\tilde{\Omega}}(X) = \mathcal{P}_{\Omega\cup\tilde{\Omega}}(M).$$

output Completed matrix $\hat{M}$.

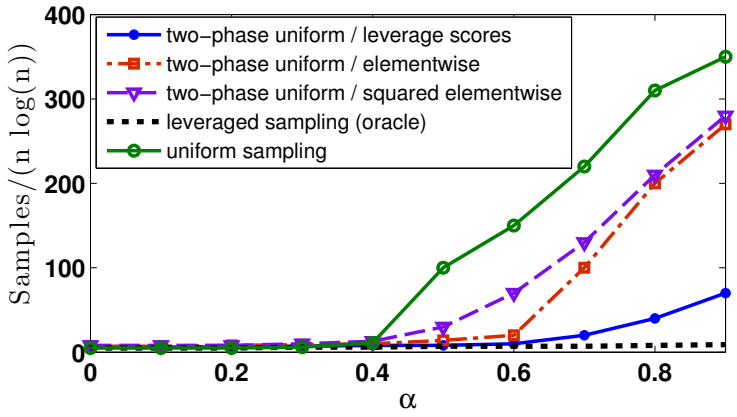


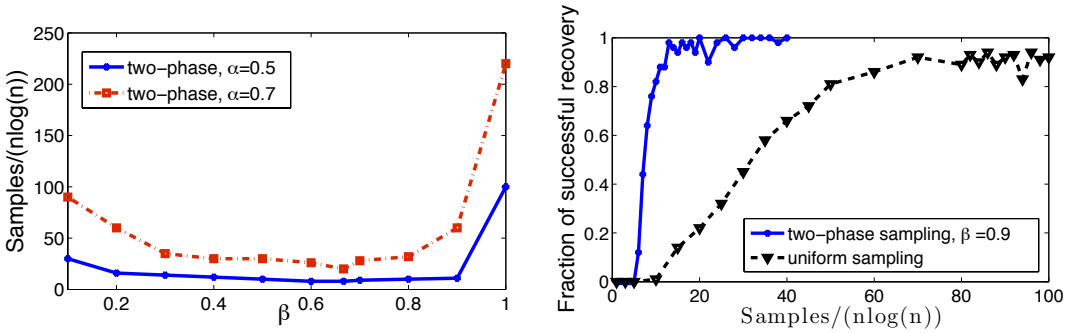
Figure 1: Performance of Algorithm 1 for power-law matrices: We consider rank-5 matrices of the form $M = DUV^\top D$, where elements of the matrices U and V are generated independently from a Gaussian distribution $\mathcal{N}(0,1)$ and D is a diagonal matrix with $D_{ii} = \frac{1}{i^\alpha}$. Higher values of α correspond to more non-uniform leverage scores and less incoherent matrices. The above simulations are run with two-phase parameter $\beta = 2/3$. Leveraged sampling (3) gives the best results of successful recovery using roughly $10n\log(n)$ samples for all values of α in accordance with Theorem 2. Surprisingly, sampling according to (6) with estimated leverage scores has almost the same sample complexity for $\alpha \leq 0.7$. Uniform sampling and sampling proportional to element and element squared perform well for low values of α, but their performance degrades quickly for $\alpha > 0.6$.

M is incoherent, then already the algorithm will recover M if $m_1 = \Theta(nr\log^2(2n))$. On the other hand, if M is *highly* coherent, having almost all energy concentrated on just a few elements, then the estimated leverage scores (6) from uniform sampling in the first step will be poor and hence the recovery algorithm suffers. Between these two extremes, there is reason to believe that the two-phase sampling procedure will provide a better estimate to the underlying matrix than if all m elements were sampled uniformly. Indeed, numerical experiments suggest that the two-phase procedure can indeed significantly outperform uniform sampling for completing coherent matrices.

4.1 Numerical Experiments

We now study the performance of the two-phase sampling procedure outlined in Algorithm 1 through numerical experiments. For this, we consider rank-5 matrices of size 500×500 of the form $M = DUV^\top D$, where the elements of the matrices U and V are i.i.d. Gaussian $\mathcal{N}(0,1)$ and D is a diagonal matrix with power-law decay, $D_{ii} = i^{-\alpha}, 1 \leq i \leq 500$. We refer to such constructions as *power-law* matrices. The parameter α adjusts the leverage scores (and hence the coherence level) of M with $\alpha = 0$ being maximal incoherence $\mu_0 = \Theta(1)$ and $\alpha = 1$ corresponding to maximal coherence $\mu_0 = \Theta(n)$.

Figure 1 plots the number of samples required for successful recovery (y-axis) for different values of α (x-axis) and $\beta = 2/3$ using Algorithm 1 with the initial samples Ω



(a) (b)

Figure 2: We consider power-law matrices with parameter $\alpha = 0.5$ and $\alpha = 0.7$. (a): This plot shows that Algorithm 1 successfully recovers coherent low-rank matrices with fewest samples ($\approx 10n\log(n)$) when the proportion of initial samples drawn from the uniform distribution is in the range $\beta \in [0.5, 0.8]$. In particular, the sample complexity is significantly lower than that for uniform sampling ($\beta = 1$). Note the x-axis starts at 0.1. (b): Even by drawing 90% of the samples uniformly and using the estimated leverage scores to sample the remaining 10% samples, one observes a marked improvement in the rate of recovery.

taken uniformly at random. Successful recovery is defined as when at least 95% of trials have relative errors in the Frobenius norm $\|M - \hat{M}\|_F / \|M\|_F$ not exceeding 0.01. To put the results in perspective, we plot it in Figure 1 against the performance of pure uniform sampling, as well as other popular sampling distributions from the matrix sparsification literature (Achlioptas and McSherry, 2007; Achlioptas et al., 2013; Arora et al., 2006; Drineas and Zouzias, 2011), namely, in step 2 of the algorithm, sampling proportional to element ($\tilde{p}_{ij} \propto |\tilde{M}_{ij}|$) and sampling proportional to element squared ($\tilde{p}_{ij} \propto \tilde{M}_{ij}^2$), as opposed to sampling from the distribution (6). In all cases, the estimated matrix $\tilde{M}$ is constructed from the rank-r SVD of $\mathcal{P}_\Omega(M)$, $\tilde{M} = \tilde{U}\tilde{\Sigma}\tilde{V}^\top$. Performance of nuclear norm minimization using samples generated according to the "oracle" distribution (3) serves as baseline for the best possible recovery, as theoretically justified by Theorem 2. We use the Augmented Lagrangian Method (ALM) based solver in Lin et al. (2009) to solve the convex optimization program (1).

Figure 1 suggests that the two-phase algorithm performs comparably to the theoretically optimal leverage scores-based distribution (3), despite not having access to the underlying leverage scores, in the regime of mild to moderate coherence. While the element-wise sampling strategies perform comparably for low values of α, the number of samples for successful recovery increases quickly for $\alpha > 0.6$. Completion from purely uniformly sampled elements requires significantly more samples at higher values of α.

Choosing β: Recall that the parameter β in Algorithm 1 is the fraction of uniform samples used to estimate the leverage scores. Figure 2(a) plots the number of samples required for successful recovery (y-axis) as β (x-axis) varies from 0 to 1 for different values of α. Setting $\beta = 1$ reduces to purely uniform sampling, and for small values of β, the leverage scores estimated in (6) will be far from the actual leverage scores. Then, as expected, the

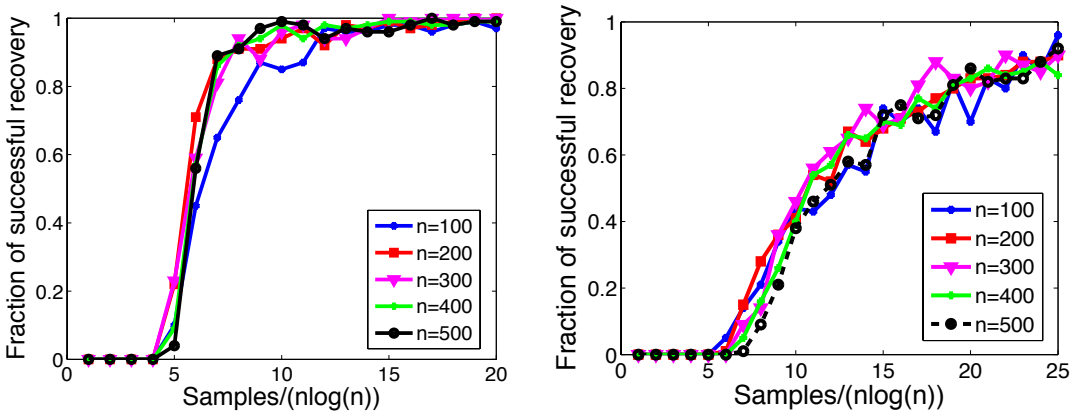


Figure 3: Scaling of sample complexity of Algorithm 1 with n. We consider power-law matrices with $\alpha = 0.5$ in plot (a) and 0.7 in plot (b). We set $\beta = 2/3$ for this set of simulations. The plots suggest that the sample complexity of Algorithm 1 scales roughly as $\Theta(n \log(n))$.

sample complexity goes up for β near 0 and $\beta = 1$. We find the algorithm performs well for a wide range of β, and setting $\beta \approx 2/3$ results in the lowest sample complexity. Surprisingly, even taking $\beta = 0.9$ as opposed to pure uniform sampling $\beta = 1$ results in a significant decrease in the sample complexity; see Figure 2(b) for more details. That is, even budgeting just a small fraction of samples to be drawn from the estimated leverage scores can significantly improve the success rate in low-rank matrix recovery as long as the underlying matrix is not completely coherent. In applications like collaborative filtering, this would imply that incentivizing just a small fraction of users to rate a few selected movies according to the estimated leverage score distribution obtained by previous samples has the potential to greatly improve the quality of the recovered matrix of preferences.

In Figure 3 we compare the performance of the two-phase algorithm for different values of the matrix dimension n, and notice for each n a phase transition occurring at $\Theta(n \log(n))$ samples. In Figure 4 we consider the scenario where the samples are noisy and compare the performance of Algorithm 1 to uniform sampling and the theoretically-optimal leveraged sampling from Theorem 2. Specifically we assume that the samples are generated from $M + Z$ where Z is a Gaussian noise matrix. We consider two values for the noise $\sigma \stackrel{\text{def}}{=} \|Z\|_F / \|M\|_F$: $\sigma = 0.1$ and $\sigma = 0.2$. The figures plot relative error in Frobenius norm (y-axis), vs total number of samples m (x-axis). These plots demonstrate the robustness of the algorithm to noise and once again show that sampling with estimated leverage scores can be as good as sampling with exact leverage scores for matrix recovery using nuclear norm minimization for $\alpha \leq 0.7$.

The empirical results in this section demonstrate the advantage of the two-phase algorithm over uniform sampling. It is an interesting future problem to provide rigorous analysis on the sample complexity of the algorithm. We note that there is an $\Omega(n^2)$ lower bound on

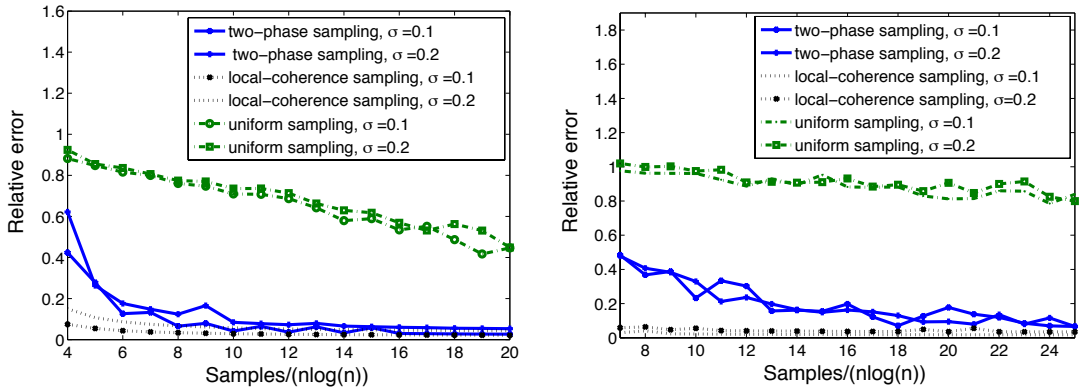


Figure 4: Performance of Algorithm 1 with noisy samples: We consider power-law matrices (with $\alpha = 0.5$ in plot (a) and $\alpha = 0.7$ in plot (b)), perturbed by a Gaussian noise matrix Z with $\|Z\|_F/\|M\|_F = \sigma$. The plots consider two different noise levels, $\sigma = 0.1$ and $\sigma = 0.2$. We compare two-phase sampling (Algorithm 1) with $\beta = 2/3$, sampling from the exact leverage scores, and uniform sampling. Algorithm 1 has relative error almost as low as the leveraged sampling without requiring any a priori knowledge of the low-rank matrix, while uniform sampling suffers dramatically.

the sample complexity for algorithms using passive sampling when the underlying matrix is maximally coherent (Krishnamurthy and Singh, 2014).

5. Weighted Nuclear Norm Minimization

Theorem 2 suggests that the performance of nuclear norm minimization will be better if the set of observed elements is aligned with the leverage scores of the matrix. Interestingly, Theorem 2 can also be used in a reverse way: *one may adjust the leverage scores to align with a given set of observed elements*. Here we demonstrate an application of this idea in quantifying the benefit of *weighted* nuclear norm minimization when the revealed elements are distributed non-uniformly.

Suppose the underlying matrix of interest is incoherent. In many applications, we do not have the freedom to choose which elements to observe. Instead, the revealed elements are *given* to us, and distributed non-uniformly among the rows and columns. As observed in Srebro and Salakhutdinov (2010), standard unweighted nuclear norm minimization (1) is inefficient in this setting. They propose to instead use weighted nuclear norm minimization for low-rank matrix completion:

$$\begin{aligned} \hat{X} = \arg \min_{X \in \mathbb{R}^{n_1 \times n_2}} \quad & \|RXC\|_* \\ \text{s.t.} \quad & X_{ij} = M_{ij}, \text{ for } (i,j) \in \Omega, \end{aligned} \tag{7}$$

where $R = \text{diag}(R_1, R_2, \ldots, R_{n_1}) \in \mathbb{R}^{n_1 \times n_1}$ and $C = \text{diag}(C_1, C_2, \ldots, C_{n_2}) \in \mathbb{R}^{n_2 \times n_2}$ are user-specified diagonal weight matrices with positive diagonal elements.

We now provide a theoretical guarantee for this method, and quantify its advantage over unweighted nuclear norm minimization. Our analysis is based on the observation that weighted nuclear norm minimization can be viewed as a way of scaling the rows and columns of the underlying matrix so that its leverage scores are adjusted to reflect the given non-uniform sampling distributions. Suppose $M \in \mathbb{R}^{n_1 \times n_2}$ has rank r and satisfies the standard incoherence condition $\max_{i,j} \{\mu_i(M), \nu_j(M)\} \le \mu_0$. Let $\lfloor x \rfloor$ denote the largest integer not exceeding x. Under this setting, we can apply Theorem 2 to establish the following:

Theorem 7 *Without loss of generality, assume $R_1 \le R_2 \le \cdots \le R_{n_1}$ and $C_1 \le C_2 \le \cdots \le C_{n_2}$. There exists a universal constant c_0 such that M is the unique optimum to (7) with probability at least $1 - 5(n_1+n_2)^{-10}$ provided that for all i, j, $p_{ij} \ge \frac{1}{\min\{n_1,n_2\}^{10}}$ and*

$$p_{ij} \ge c_0 \left(\frac{R_i^2}{\sum_{i'=1}^{\lfloor n_1/(\mu_0 r) \rfloor} R_{i'}^2} + \frac{C_j^2}{\sum_{j'=1}^{\lfloor n_2/(\mu_0 r) \rfloor} C_{j'}^2} \right) \log^2 n. \tag{8}$$

This theorem is proved by drawing a connection between the weighted nuclear norm formulation (7) and the leverage scores (2) of the target matrix. Define the scaled matrix $\bar{M} := RMC$. Observe that the weighted program (7) is equivalent to first solving the following *unweighted* problem with scaled observations

$$\begin{aligned} \bar{X} = \arg\min_X \quad & \|X\|_* \\ \text{s.t.} \quad & X_{ij} = \bar{M}_{ij}, \text{ for } (i,j) \in \Omega, \end{aligned} \tag{9}$$

and then rescaling the solution $\bar{X}$ to return $\hat{X} = R^{-1}\bar{X}C^{-1}$. In other words, through the use of the weighted nuclear norm, we convert the problem of completing M to that of completing the scaled matrix $\bar{M}$. This leads to the following observation, which underlines the proof of Theorem 7:

> *If we can choose the weights R and C in such a way that the leverage scores of scaled matrix $\bar{M}$, denoted as $\bar{\mu}_i := \mu_i(\bar{M}), \bar{\nu}_j := \nu_i(\bar{M}), i, j \in [n]$, are aligned with the given non-uniform observations in a way that roughly satisfies the relation* (3)*, then we gain in sample complexity compared to the unweighted approach.*

We now quantify this observation more precisely for a particular class of matrix completion problems.

5.1 Comparison to Unweighted Nuclear Norm.

Assume for simplicity $n_1 = n_2 = n$ and $n/(\mu_0 r)$ is an integer. Suppose the sampling probabilities have a product form: $p_{ij} = p_i^{\mathrm{r}} p_j^{\mathrm{c}}$, with $p_1^{\mathrm{r}} \le p_2^{\mathrm{r}} \le \cdots \le p_n^{\mathrm{r}}$ and $p_1^{\mathrm{c}} \le p_2^{\mathrm{c}} \le \cdots \le p_n^{\mathrm{c}}$. If we choose $R_i = \sqrt{\frac{1}{n} p_i^{\mathrm{r}} \sum_{j'} p_{j'}^{\mathrm{c}}}$ and $C_j = \sqrt{\frac{1}{n} p_j^{\mathrm{c}} \sum_{i'} p_{i'}^{\mathrm{r}}}$—which is suggested by the condition (8)—Theorem 7 asserts that the following set of conditions are sufficient for recovery of M with high probability:

$$p_j^{\mathrm{c}} \cdot \left(\frac{\mu_0 r}{n} \sum_{i=1}^{n/(\mu_0 r)} p_i^{\mathrm{r}} \right) \gtrsim \frac{\mu_0 r}{n} \log^2 n, \ \forall j; \qquad p_i^{\mathrm{r}} \cdot \left(\frac{\mu_0 r}{n} \sum_{j=1}^{n/(\mu_0 r)} p_j^{\mathrm{c}} \right) \gtrsim \frac{\mu_0 r}{n} \log^2 n, \ \forall i. \tag{10}$$

We can compare the above condition to the condition for the unweighted approach: by Theorem 2, the unweighted nuclear norm minimization formulation (1) recovers M if

$$p_i^{\mathrm{r}} \cdot p_j^{\mathrm{c}} \gtrsim \frac{\mu_0 r}{n} \log^2 n, \quad \forall i, j. \tag{11}$$

Therefore, the weighted nuclear norm approach succeeds under less restrictive conditions: the condition (11) for the unweighted approach requires a lower bound on *minimum* sampling probability over the rows and columns, whereas the condition (10) for the weighted approach involves the *average* sampling probability of the $n/(\mu_0 r)$ least sampled rows/columns. This benefit is most significant precisely when the observed samples are very non-uniformly distributed.

We provide a concrete example of the gain of weighting in Section E.

Our theoretical results are consistent with the empirical study in Srebro and Salakhutdinov (2010); Foygel et al. (2011), which demonstrate the advantage of the weighted approach with the weights R and C chosen as above (using the empirical sampling distribution). We remark that Theorem 7 is the first exact recovery guarantee for weighted nuclear norm minimization. It provides a theoretical explanation, complementary to those in Srebro and Salakhutdinov (2010); Foygel et al. (2011); Negahban and Wainwright (2012), for why the weighted approach is advantageous over the unweighted approach for non-uniform observations. It also serves as a testament to the power of Theorem 2 as a general result on the relationship between sampling and the coherence/leverage score structure of a matrix.

In Theorem 7 and the discussion above we assume the underlying matrix M is incoherent. Clearly, one can still use the weighted nuclear norm approach when M is coherent: as long as the weights are chosen such that the leverage scores of the scaled matrix $\bar{M}$ are aligned with the distributions of the revealed entries, Theorem 2 can be applied and we expect improvements of the recovery performance using the weighted approach. How to choose the weights in this setting, and how it affects the performance, are left to future work.

6. Conclusion

In this paper we study the problem of matrix completion with no assumptions on the incoherence of the underlying matrix. We show that if the sampling of entries suitably depends on leverage scores of the matrix, then it can be recovered from $O(nr \log^2(n))$ entries using nuclear norm minimization. We further establish the necessity of leverage score sampling within the class of location invariant sampling distributions. Based on these results, we present a new two-phase sampling algorithm which does not require knowledge of underlying structure of the matrix and provide simulation results to verify its performance. As a corollary of our main theorem, we provide exact recovery guarantees for the weighted nuclear norm minimization approach when the observed entries are given and distributed non-uniformly.

It is an interesting open problem to provide rigorous theoretical analysis of the number of samples needed by the two-phase sampling algorithm. It is also of interest to develop and analyze algorithms that sample with more stages and iteratively improve the leverage score estimates. More generally, it is useful to develop and study other methods for estimating/adjusting the leverage scores and tuning the sampling procedure. Extending the

results in this paper to other low-rank recovery settings and applications will be of great value.

Acknowledgments

We would like to thank Petros Drineas, Michael Mahoney and Aarti Singh for helpful discussions, and the anonymous reviewers for their insightful comments and suggestions. Y. Chen was supported by NSF grant CIF-31712-23800 and ONR MURI grant N00014-11-1-0688. R. Ward was supported in part by an NSF CAREER award, AFOSR Young Investigator Program award, and ONR Grant N00014-12-1-0743. S. Sanghavi would like to acknowledge NSF grants 1302435, 1320175 and 0954059 for supporting this work.

Appendix A. Proof of Theorem 2

We prove our main result Theorem 2 in this section. The overall outline of the proof is a standard convex duality argument. The main difference in establishing our results is that, while other proofs relied on bounding the ℓ_∞ norm of certain random matrices, we instead bound the weighted $\ell_{\infty,2}$, norm (to be defined).

The high level road map of the proof is a standard one: by convex analysis, to show that M is the unique optimal solution to (1), it suffices to construct a *dual certificate* Y obeying certain sub-gradient optimality conditions. One of the conditions requires the spectral norm $\|Y\|$ to be small. Previous work bounds $\|Y\|$ by the the ℓ_∞ norm $\|Y'\|_\infty := \sum_{i,j} \left|Y'_{ij}\right|$ of a certain matrix Y', which gives rise to the standard and joint incoherence conditions involving uniform bounds by μ_0 and μ_{str}. Here, we derive a new bound using the weighted $\ell_{\infty,2}$ norm of Y', which is the maximum of the weighted row and column norms of Y'. These bounds lead to a tighter bound of $\|Y\|$ and hence less restrictive conditions for matrix completion.

We now turn to the details. To simplify the notion, we prove the results for square matrices ($n_1 = n_2 = n$). The results for non-square matrices are proved in exactly the same fashion. In the sequel by *with high probability* (*w.h.p.*) we mean with probability at least $1 - n^{-20}$. The proof below involves no more than $5n^{10}$ random events, each of which will be shown to hold with high probability. It follows from the union bound that all the events simultaneously hold with probability at least $1 - 5n^{-10}$, which is the success probability in the statement of Theorem 2.

A few additional notations are needed. We drop the dependence of $\mu_i(M)$ and $\nu_j(M)$ on M and simply use μ_i and ν_j. We use c and its derivatives (c', c_0, etc.) for universal positive constants, which may differ from place to place. The inner product between two matrices is given by $\langle Y, Z\rangle = \mathrm{trace}(Y^\top Z)$. Recall that U and V are the left and right singular vectors of the underlying matrix M. We need several standard projection operators for matrices. The projections P_T and $P_{T^\perp}$ are given by

$$P_T(Z) := UU^\top Z + ZVV^\top - UU^\top VZZ^\top$$

and $P_{T^\perp}(Z) := Z - P_T(Z)$. $P_\Omega(Z)$ is the matrix with $(P_\Omega(Z))_{ij} = Z_{ij}$ if $(i,j) \in \Omega$ and zero otherwise, and $P_{\Omega^c}(Z) := Z - P_\Omega(Z)$. As usual, $\|z\|_2$ is the ℓ_2 norm of the vector z, and $\|Z\|_F$ and $\|Z\|$ are the Frobenius norm and spectral norm of the matrix

Z, respectively. For a linear operator R on matrices, its operator norm is defined as $\|R\|_{op} = \sup_{X \in \mathbb{R}^{n\times n}} \|R(X)\|_F / \|X\|_F$. For each $1 \le i, j \le n$, we define the random variable $\delta_{ij} := \mathbb{I}((i,j) \in \Omega)$, where $\mathbb{I}(\cdot)$ is the indicator function. The matrix operator $R_\Omega : \mathbb{R}^{n\times n} \mapsto \mathbb{R}^{n\times n}$ is defined as

$$R_\Omega(Z) = \sum_{i,j} \frac{1}{p_{ij}} \delta_{ij} \left\langle e_i e_j^\top, Z \right\rangle e_i e_j^\top. \tag{12}$$

A.1 Optimality Condition

Following our proof road map, we now state a sufficient condition for M to be the unique optimal solution to the optimization problem (1). This is the content of Proposition 8 below (proved in Section A.7 to follow).

Proposition 8 *Suppose $p_{ij} \ge \frac{1}{n^{10}}$. The matrix M is the unique optimal solution to* (1) *if the following conditions hold.*

1. $\|P_T R_\Omega P_T - P_T\|_{op} \le \frac{1}{2}$.
2. *There exists a dual certificate $Y \in \mathbb{R}^{n\times n}$ which satisfies $P_\Omega(Y) = Y$ and*
   (a) $\left\|P_T(Y) - UV^\top\right\|_F \le \frac{1}{4n^5}$,
   (b) $\|P_{T^\perp}(Y)\| \le \frac{1}{2}$.

A.2 Validating the Optimality Condition

We begin by proving that Condition 1 in Proposition 8 is satisfied under the conditions of Theorem 2. This is done in the following lemma, which is proved in Section A.8 to follow. The lemma shows that R_Ω is close to the identity operator on T.

Lemma 9 *If $p_{ij} \ge \min\{c_0 \frac{(\mu_i+\nu_j)r}{n} \log n, 1\}$ for all (i,j) and a sufficiently large c_0, then w.h.p.*

$$\|P_T R_\Omega P_T - P_T\|_{op} \le \frac{1}{2}. \tag{13}$$

A.3 Constructing the Dual Certificate

It remains to construct a matrix Y (the dual certificate) that satisfies Condition 2 in Proposition 8. We do this using the golfing scheme (Gross, 2011; Candès et al., 2011). Set $k_0 := 20 \log n$. For each $k = 1, \ldots, k_0$, let $\Omega_k \subseteq \mathbb{R}^{n\times n}$ be a random set of matrix elements such that for each (i,j), $\mathbb{P}[(i,j) \in \Omega_k] = q_{ij} := 1 - (1-p_{ij})^{1/k_0}$, independently of all others. We may assume that the set Ω of observed elements is generated as $\Omega = \bigcup_{k=1}^{k_0} \Omega_k$, which is equivalent to the original Bernoulli sampling model. Let $W_0 := 0$ and for $k = 1, \ldots, k_0$,

$$W_k := W_{k-1} + R_{\Omega_k} P_T (UV^\top - P_T W_{k-1}), \tag{14}$$

where the operator R_{Ω_k} is given by

$$R_{\Omega_k}(Z) = \sum_{i,j} \frac{1}{q_{ij}} \mathbb{I}((i,j) \in \Omega_k) \left\langle e_i e_j^\top, Z \right\rangle e_i e_j^\top.$$

The dual certificate is given $Y := W_{k_0}$. Clearly $P_\Omega(Y) = Y$ by construction. The proof of Theorem 2 is completed if we show that under the condition in the theorem, Y satisfies Conditions 2(a) and 2(b) in Proposition 8 w.h.p.

A.4 Concentration Properties

The key step in our proof is to show that Y satisfies Condition 2(b) in Proposition 8, i.e., we need to bound $\|P_{T^\perp}(Y)\|$. Here our proof departs from existing ones, as we establish concentration bounds on this quantity in terms of (an appropriately weighted version of) the $\ell_{\infty,2}$ norm, which we now define. The $\mu(\infty,2)$-norm of a matrix $Z \in \mathbb{R}^{n\times n}$ is defined as

$$\|Z\|_{\mu(\infty,2)} := \max\left\{ \max_i \sqrt{\frac{n}{\mu_i r}\sum_b Z_{ib}^2}, \max_j \sqrt{\frac{n}{\nu_j r}\sum_a Z_{aj}^2} \right\},$$

which is the maximum of the weighted column and row norms of Z. We also need the $\mu(\infty)$-norm of Z, which is a weighted version of the matrix ℓ_∞ norm. This is given as

$$\|Z\|_{\mu(\infty)} := \max_{i,j} |Z_{ij}| \sqrt{\frac{n}{\mu_i r}}\sqrt{\frac{n}{\nu_j r}}.$$

which is the weighted element-wise magnitude of Z. We now state three new lemmas concerning the concentration properties of these norms. The first lemma is crucial to our proof; it bounds the spectral norm of $(R_\Omega - I)\, Z$ in terms of the $\mu(\infty,2)$ and $\mu(\infty)$ norms of Z. This obviates the intermediate lemmas required by previous approaches (Candès and Tao, 2010; Gross, 2011; Recht, 2011; Keshavan et al., 2010) which use the ℓ_∞ norm of Z.

Lemma 10 *Suppose Z is a fixed $n \times n$ matrix. For some universal constant $c > 1$, we have w.h.p.*

$$\|(R_\Omega - I)\, Z\| \le c\left(\max_{i,j}\left|\frac{Z_{ij}}{p_{ij}}\right| \log n + \sqrt{\max\left\{ \max_i \sum_{j=1}^n \frac{Z_{ij}^2}{p_{ij}}, \max_j \sum_{i=1}^n \frac{Z_{ij}^2}{p_{ij}} \right\} \log n} \right).$$

If $p_{ij} \ge \min\{c_0 \frac{(\mu_i+\nu_j)r}{n} \log n, 1\}$ for all (i,j) and a sufficiently large constant c_0, then we further have w.h.p.

$$\|(R_\Omega - I)\, Z\| \le \frac{c}{\sqrt{c_0}} \left(\|Z\|_{\mu(\infty)} + \|Z\|_{\mu(\infty,2)} \right).$$

The next two lemmas further control the $\mu(\infty,2)$ and $\mu(\infty)$ norms of a matrix after certain random transformation.

Lemma 11 *Suppose Z is a fixed $n \times n$ matrix. If $p_{ij} \ge \min\{c_0 \frac{(\mu_i+\nu_j)r}{n} \log n, 1\}$ for all i, j and a sufficiently large constant c_0, then w.h.p.*

$$\|(P_T R_\Omega - P_T)Z\|_{\mu(\infty,2)} \le \frac{1}{2}\left(\|Z\|_{\mu(\infty)} + \|Z\|_{\mu(\infty,2)} \right)$$

Lemma 12 *Suppose Z is a fixed $n \times n$ matrix. If $p_{ij} \geq \min\{c_0 \frac{(\mu_i+\nu_j)r}{n} \log n, 1\}$ for all i, j and a sufficiently large constant c_0, then w.h.p.*

$$\|(P_T R_\Omega - P_T)\, Z\|_{\mu(\infty)} \leq \frac{1}{2} \|Z\|_{\mu(\infty)}\,.$$

We prove Lemmas 10–12 in Section A.8. Equipped with the three lemmas above, we are now ready to validate that Y satisfies Condition 2 in Proposition 8.

A.5 Validating Condition 2(a)

Set $\Delta_k = UV^\top - P_T(W_k)$ for $k = 1, \ldots, k_0$; note that $\Delta_{k_0} = UV^\top - P_T(Y)$. By definition of W_k, we have

$$\Delta_k = (P_T - P_T R_{\Omega_k} P_T)\, \Delta_{k-1}. \tag{15}$$

Note that Ω_k is independent of Δ_{k-1} and $q_{ij} \geq p_{ij}/k_0 \geq c_0'(\mu_i + \nu_j) r \log(n)/n$ under the condition in Theorem 2. Applying Lemma 9 with Ω replaced by Ω_k , we obtain that w.h.p.

$$\|\Delta_k\|_F \leq \|P_T - P_T R_{\Omega_k} P_T\|\, \|\Delta_{k-1}\|_F \leq \frac{1}{2} \|\Delta_{k-1}\|_F\,.$$

Applying the above inequality recursively with $k = k_0, k_0 - 1, \ldots, 1$ gives

$$\left\|P_T(Y) - UV^\top\right\|_F = \|\Delta_{k_0}\|_F \leq \left(\frac{1}{2}\right)^{k_0} \left\|UV^\top\right\|_F \leq \frac{1}{4n^6} \cdot \sqrt{r} \leq \frac{1}{4n^5},$$

where we use our definition of k_0 and $\left\|UV^\top\right\|_F = \sqrt{r}$ in the second inequality.

A.6 Validating Condition 2(b)

By definition, Y can be rewritten as $Y = \sum_{k=1}^{k_0} R_{\Omega_k} P_T \Delta_{k-1}$. It follows that

$$\|P_{T^\perp}(Y)\| = \left\|P_{T^\perp} \sum_{k=1}^{k_0} (R_{\Omega_k} P_T - P_T)\, \Delta_{k-1}\right\| \leq \sum_{k=1}^{k_0} \|(R_{\Omega_k} - I)\, \Delta_{k-1}\|\,.$$

We apply Lemma 10 with Ω replaced by Ω_k to each summand in the last RHS to obtain w.h.p.

$$\|P_{T^\perp}(Y)\| \leq \frac{c}{\sqrt{c_0}} \sum_{k=1}^{k_0} \|\Delta_{k-1}\|_{\mu(\infty)} + \frac{c}{\sqrt{c_0}} \sum_{k=1}^{k_0} \|\Delta_{k-1}\|_{\mu(\infty,2)}\,. \tag{16}$$

We bound each summand in the last RHS. Applying $(k-1)$ times (15) and Lemma 12 (with Ω replaced by Ω_k), we have w.h.p.

$$\|\Delta_{k-1}\|_{\mu(\infty)} = \left\|\left(P_T - P_T R_{\Omega_{k-1}} P_T\right) \Delta_{k-2}\right\|_{\mu(\infty)} \leq \left(\frac{1}{2}\right)^{k-1} \left\|UV^\top\right\|_{\mu(\infty)}\,.$$

for each k. Similarly, repeatedly applying (15), Lemma 11 and the inequality we just proved above, we obtain w.h.p.

$$\begin{aligned}
&\|\Delta_{k-1}\|_{\mu(\infty,2)} && (17)\\
=&\left\|\left(P_T - P_T R_{\Omega_{k-1}} P_T\right)\Delta_{k-2}\right\|_{\mu(\infty,2)} && (18)\\
\leq&\frac{1}{2}\|\Delta_{k-2}\|_{\mu(\infty)} + \frac{1}{2}\|\Delta_{k-2}\|_{\mu(\infty,2)} && (19)\\
\leq&\left(\frac{1}{2}\right)^{k-1}\left\|UV^\top\right\|_{\mu(\infty)} + \frac{1}{2}\|\Delta_{k-2}\|_{\mu(\infty,2)} && (20)\\
\leq&k\left(\frac{1}{2}\right)^{k-1}\left\|UV^\top\right\|_{\mu(\infty)} + \left(\frac{1}{2}\right)^{k-1}\|UV\|_{\mu(\infty,2)}\,. && (21)
\end{aligned}$$

It follows that w.h.p.

$$\begin{aligned}
\|P_{T^\perp}(Y)\| &\leq \frac{c}{\sqrt{c_0}}\sum_{k=1}^{k_0}(k+1)\left(\frac{1}{2}\right)^{k-1}\left\|UV^\top\right\|_{\mu(\infty)} + \frac{c}{\sqrt{c_0}}\sum_{k=1}^{k_0}\left(\frac{1}{2}\right)^{k-1}\left\|UV^\top\right\|_{\mu(\infty,2)} && (22)\\
&\leq \frac{6c}{\sqrt{c_0}}\left\|UV^\top\right\|_{\mu(\infty)} + \frac{2c}{\sqrt{c_0}}\left\|UV^\top\right\|_{\mu(\infty,2)}\,. && (23)
\end{aligned}$$

Note that for all (i,j), we have $\left|\left(UV^\top\right)_{ij}\right| = \left|e_i^\top UV^\top e_j\right| \leq \sqrt{\frac{\mu_i r}{n}}\sqrt{\frac{\nu_j r}{n}}$, $\left\|e_i^\top UV^\top\right\|_2 = \sqrt{\frac{\mu_i r}{n}}$ and $\left\|UV^\top e_j\right\|_2 = \sqrt{\frac{\nu_j r}{n}}$. Hence $\left\|UV^\top\right\|_{\mu(\infty)} \leq 1$ and $\left\|UV^\top\right\|_{\mu(\infty,2)} = 1$. We conclude that

$$\|P_{T^\perp}(Y)\| \leq \frac{6c}{\sqrt{c_0}} + \frac{2c}{\sqrt{c_0}} \leq \frac{1}{2}$$

provided that the constant c_0 in Theorem 2 is sufficiently large. This completes the proof of Theorem 2.

A.7 Proof of Proposition 8 (Optimality Condition)

Proof Consider any feasible solution X to (1) with $P_\Omega(X) = P_\Omega(M)$. Let G be an $n \times n$ matrix which satisfies $\|P_{T^\perp}G\| = 1$, and $\langle P_{T^\perp}G, P_{T^\perp}(X-M)\rangle = \|P_{T^\perp}(X-M)\|_*$. Such G always exists by duality between the nuclear norm and spectral norm. Because $UV^\top + P_{T^\perp}G$ is a sub-gradient of the function $f(Z) = \|Z\|_*$ at $Z = M$, we have

$$\|X\|_* - \|M\|_* \geq \left\langle UV^\top + P_{T^\perp}G, X - M\right\rangle. \tag{24}$$

But $\langle Y, X-M\rangle = \langle P_\Omega(Y), P_\Omega(X-M)\rangle = 0$ since $P_\Omega(Y) = Y$. It follows that

$$\begin{aligned}
&\|X\|_* - \|M\|_* \geq \left\langle UV^\top + P_{T^\perp}G - Y, X - M\right\rangle\\
&= \|P_{T^\perp}(X-M)\|_* + \left\langle UV^\top - P_T Y, X - M\right\rangle - \langle P_{T^\perp}Y, X - M\rangle\\
&\geq \|P_{T^\perp}(X-M)\|_* - \left\|UV^\top - P_T Y\right\|_F \|P_T(X-M)\|_F - \|P_{T^\perp}Y\|\,\|P_{T^\perp}(X-M)\|_*\\
&\geq \frac{1}{2}\|P_{T^\perp}(X-M)\|_* - \frac{1}{4n^5}\|P_T(X-M)\|_F\,,
\end{aligned}$$

where in the last inequality we use conditions $2a$ and $2b$ in the proposition. Using Lemma 13 below, we obtain

$$\|X\|_* - \|M\|_* \geq \frac{1}{2}\|P_{T^\perp}(X-M)\|_* - \frac{1}{4n^5}\cdot\sqrt{2}n^5\|P_{T^\perp}(X-M)\|_* > \frac{1}{8}\|P_{T^\perp}(X-M)\|_*.$$

The RHS is strictly positive for all X with $P_\Omega(X-M)=0$ and $X \neq M$. Otherwise we must have $P_T(X-M) = X-M$ and $P_T P_\Omega P_T(X-M) = 0$, contradicting the assumption $\|P_T R_\Omega P_T - P_T\|_{op} \leq \frac{1}{2}$. This proves that M is the unique optimum. ■

Lemma 13 *If $p_{ij} \geq \frac{1}{n^{10}}$ for all (i,j) and $\|P_T R_\Omega P_T - P_T\|_{op} \leq \frac{1}{2}$, then we have*

$$\|P_T Z\|_F \leq \sqrt{2}n^5\|P_{T^\perp}(Z)\|_*, \forall Z \in \{Z' : P_\Omega(Z') = 0\}. \tag{25}$$

Proof Define the operator $R_\Omega^{1/2} : \mathbb{R}^{n\times n} \mapsto \mathbb{R}^{n\times n}$ by

$$R_\Omega^{1/2}(Z) := \sum_{i,j}\frac{1}{\sqrt{p_{ij}}}\delta_{ij}\left\langle e_i e_j^\top, Z\right\rangle e_i e_j^\top.$$

Note that $R_\Omega^{1/2}$ is self-adjoint and satisfies $R_\Omega^{1/2}R_\Omega^{1/2} = R_\Omega$. Hence we have

$$\begin{aligned}
\left\|R_\Omega^{1/2}P_T(Z)\right\|_F &= \sqrt{\langle P_T R_\Omega P_T Z, P_T Z\rangle} \\
&= \sqrt{\langle (P_T R_\Omega P_T - P_T)Z, P_T(Z)\rangle + \langle P_T(Z), P_T(Z)\rangle} \\
&\geq \sqrt{\|P_T(Z)\|_F^2 - \|P_T R_\Omega P_T - P_T\|\,\|P_T(Z)\|_F^2} \\
&\geq \frac{1}{\sqrt{2}}\|P_T(Z)\|_F,
\end{aligned}$$

where the last inequality follows from the assumption $\|P_T R_\Omega P_T - P_T\|_{op} \leq \frac{1}{2}$. On the other hand, $P_\Omega(Z) = 0$ implies $0 = R_\Omega^{1/2}(Z) = R_\Omega^{1/2}P_T(Z) + R_\Omega^{1/2}P_{T^\perp}(Z)$ and thus

$$\left\|R_\Omega^{1/2}P_T(Z)\right\|_F = \left\|-R_\Omega^{1/2}P_{T^\perp}(Z)\right\|_F \leq \left(\max_{i,j}\frac{1}{\sqrt{p_{ij}}}\right)\|P_{T^\perp}(Z)\|_F \leq n^5\|P_{T^\perp}(Z)\|_F.$$

Combining the last two display equations gives

$$\|P_T(Z)\|_F \leq \sqrt{2}n^5\|P_{T^\perp}(Z)\|_F \leq \sqrt{2}n^5\|P_{T^\perp}(Z)\|_*.$$

■

A.8 Proof of Technical Lemmas

We prove the four technical lemmas that are used in the proof of our main theorem. The proofs use the matrix Bernstein inequality given as Theorem 16 in Section F. We also make frequent use of the following facts: for all i and j, we have $\max\left\{\frac{\mu_i r}{n}, \frac{\nu_j r}{n}\right\} \leq 1$ and

$$\frac{(\mu_i+\nu_j)r}{n} \geq \left\|P_T(e_i e_j^\top)\right\|_F^2. \tag{26}$$

We also use the shorthand $a \wedge b := \min\{a,b\}$.

A.8.1 Proof of Lemma 9

For any matrix Z, we can write

$$(P_T R_\Omega P_T - P_T)(Z) = \sum_{i,j} \left(\frac{1}{p_{ij}} \delta_{ij} - 1 \right) \left\langle e_i e_j^\top, P_T(Z) \right\rangle P_T(e_i e_j^\top) =: \sum_{i,j} \mathcal{S}_{ij}(Z).$$

Note that $\mathbb{E}\left[\mathcal{S}_{ij}\right] = 0$ and $\mathcal{S}_{ij}$'s are independent of each other. For all Z and (i,j), we have $\mathcal{S}_{ij} = 0$ if $p_{ij} = 1$. On the other hand, when $p_{ij} \geq c_0 \frac{(\mu_i + \nu_j) r \log n}{n}$, then it follows from (26) that

$$\left\| \mathcal{S}_{ij}(Z) \right\|_F \leq \frac{1}{p_{ij}} \left\| P_T(e_i e_j^\top) \right\|_F^2 \|Z\|_F \leq \max_{i,j} \left\{ \frac{1}{p_{ij}} \frac{(\mu_i + \nu_j) r}{n} \right\} \|Z\|_F \leq \frac{1}{c_0 \log n} \|Z\|_F .$$

Putting together, we have that $\|\mathcal{S}_{ij}\| \leq \frac{1}{c_0 \log n}$ under the condition of the lemma. On the other hand, we have

$$\begin{aligned}
\left\| \sum_{i,j} \mathbb{E}\left[\mathcal{S}_{ij}^2(Z) \right] \right\|_F &= \left\| \sum_{i,j} \mathbb{E}\left[\left(\frac{1}{p_{ij}} \delta_{ij} - 1 \right)^2 \left\langle e_i e_j^\top, P_T(Z) \right\rangle \left\langle e_i e_j^\top, P_T(e_i e_j^\top) \right\rangle P_T(e_i e_j^\top) \right] \right\|_F \\
&\leq \left(\max_{i,j} \frac{1 - p_{ij}}{p_{ij}} \left\| P_T(e_i e_j^\top) \right\|_F^2 \right) \left\| \sum_{i,j} \left\langle e_i e_j^\top, P_T(Z) \right\rangle P_T(e_i e_j^\top) \right\|_F \\
&\leq \max_{i,j} \left\{ \frac{1 - p_{ij}}{p_{ij}} \frac{(\mu_i + \nu_j) r}{n} \right\} \|P_T(Z)\|_F ,
\end{aligned}$$

This implies $\left\| \sum_{i,j} \mathbb{E}\left[\mathcal{S}_{ij}^2 \right] \right\| \leq \frac{1}{c_0 \log n}$ under the condition of the lemma. Applying the Matrix Bernstein inequality (Theorem 16), we obtain $\|P_T R_\Omega P_T - P_T\| = \left\| \sum_{i,j} \mathcal{S}_{ij} \right\| \leq \frac{1}{2}$ w.h.p. for sufficiently large c_0.

A.8.2 Proof of Lemma 10

We can write $(R_\Omega - I)\, Z$ as the sum of independent matrices:

$$(R_\Omega - I)\, Z = \sum_{i,j} \left(\frac{1}{p_{ij}} \delta_{ij} - 1 \right) Z_{ij} e_i e_j^\top =: \sum_{i,j} S_{ij}.$$

Note that $\mathbb{E}[S_{ij}] = 0$. For all (i,j), we have $S_{ij} = 0$ if $p_{ij} = 1$, and

$$\|S_{ij}\| \leq \frac{1}{p_{ij}} \left| Z_{ij} \right| .$$

Moreover, we have

$$\left\| \mathbb{E}\left[\sum_{i,j} S_{ij}^\top S_{ij} \right] \right\| = \left\| \sum_{i,j} Z_{ij}^2 e_i e_j^\top e_j e_i^\top \mathbb{E}\left(\frac{1}{p_{ij}} \delta_{ij} - 1 \right)^2 \right\| = \max_i \sum_{j=1}^n \frac{1 - p_{ij}}{p_{ij}} Z_{ij}^2.$$

The quantity $\left\|\mathbb{E}\left[\sum_{i,j} S_{ij}S_{ij}^\top\right]\right\|$ is bounded by $\max_j \sum_{i=1}^n (1-p_{ij})Z_{ij}^2/p_{ij}$ in a similar way. The first part of the lemma then follows from the matrix Bernstein inequality (Theorem 16). If $p_{ij} \geq 1 \wedge \frac{c_0(\mu_i+\nu_j)r\log n}{n} \geq 1 \wedge 2c_0\sqrt{\frac{\mu_i r}{n}\cdot\frac{\nu_j r}{n}}\log n$, we have for all i and j,

$$\|S_{ij}\|\log n \leq (1-\mathbb{I}(p_{ij}=1))\,\frac{1}{p_{ij}}\,|Z_{ij}|\log n \leq \frac{1}{c_0}\,\|Z\|_{\mu(\infty)}\,,$$

$$\sum_{i=1}^n \frac{1-p_{ij}}{p_{ij}} Z_{ij}^2 \log n \leq \frac{1}{c_0}\,\|Z\|_{\mu(\infty,2)}^2\,,$$

$$\sum_{j=1}^n \frac{1-p_{ij}}{p_{ij}} Z_{ij}^2 \log n \leq \frac{1}{c_0}\,\|Z\|_{\mu(\infty,2)}^2\,.$$

The second part of the lemma follows again from applying the matrix Bernstein inequality.

A.8.3 Proof of Lemma 11

Let $X = (P_T R_\Omega - P_T)\,Z$. By definition we have

$$\|X\|_{\mu(\infty,2)} = \max_{a,b}\left\{\sqrt{\frac{n}{\mu_a r}}\,\|X_{a\cdot}\|_2\,, \sqrt{\frac{n}{\nu_b r}}\,\|X_{\cdot b}\|_2\right\},$$

where $X_{a\cdot}$ and $X_{\cdot b}$ are the a-th row and b-th column of of X, respectively. We bound each term in the maximum. Observe that $\sqrt{\frac{n}{\nu_b r}}X_{\cdot b}$ can be written as the sum of independent column vectors:

$$\sqrt{\frac{n}{\nu_b r}}X_{\cdot b} = \sum_{i,j}\left(\frac{1}{p_{ij}}\delta_{ij}-1\right)Z_{ij}\left(P_T(e_i e_j^\top)e_b\right)\sqrt{\frac{n}{\nu_b r}} =: \sum_{i,j} S_{ij},$$

where $\mathbb{E}\,[S_{ij}] = 0$. To control $\|S_{ij}\|_2$ and $\left\|\mathbb{E}\left[\sum_{i,j} S_{ij}^\top S_{ij}\right]\right\|$, we first need a bound for $\left\|P_T(e_i e_j^\top)e_b\right\|_2$. If $j = b$, we have

$$\left\|P_T(e_i e_j^\top)e_b\right\|_2 = \left\|UU^\top e_i + (I-UU^\top)e_i\left\|V^\top e_b\right\|_2^2\right\|_2 \leq \sqrt{\frac{\mu_i r}{n}} + \sqrt{\frac{\nu_b r}{n}}, \tag{27}$$

where we use the triangle inequality and the definition of μ_i and ν_b . Similarly, if $j \neq b$, we have

$$\left\|P_T(e_i e_j^\top)e_b\right\|_2 = \left\|(I-UU^\top)e_i e_j^\top VV^\top e_b\right\|_2 \leq \left|e_j^\top VV^\top e_b\right|. \tag{28}$$

Now note that $\|S_{ij}\|_2 \leq (1-\mathbb{I}(p_{ij}=1))\,\frac{1}{p_{ij}}\,|Z_{ij}|\,\sqrt{\frac{n}{\nu_b r}}\left\|P_T(e_i e_j^\top)e_b\right\|_2$. Using the bounds (27) and (28), we obtain that for $j = b$,

$$\begin{aligned}\|S_{ij}\|_2 &\leq (1-\mathbb{I}(p_{ij}=1))\,\frac{1}{p_{ib}}\,|Z_{ib}|\,\sqrt{\frac{n}{\nu_b r}}\cdot\left(\sqrt{\frac{\mu_i r}{n}}+\frac{\nu_b r}{n}\right)\\ &\leq \frac{2}{c_0\sqrt{\frac{\mu_i r\nu_b r}{n^2}}\log n}\,|Z_{ib}| \leq \frac{2}{c_0\log n}\,\|Z\|_{\mu(\infty)}\,,\end{aligned}$$

where we use $p_{ib} \geq 1 \wedge \frac{c_0 \mu_i r \log n}{n}$ and $p_{ib} \geq 1 \wedge c_0 \sqrt{\frac{\mu_i r}{n} \frac{\nu_b r}{n}} \log n$ in the second inequality. For $j \neq b$, we have

$$\|S_{ij}\|_2 \leq (1 - \mathbb{I}(p_{ij} = 1)) \frac{1}{p_{ij}} |Z_{ij}| \sqrt{\frac{n}{\nu_b r}} \cdot \sqrt{\frac{\nu_j r}{n}} \sqrt{\frac{\nu_b r}{n}} \leq \frac{2}{c_0 \log n} \|Z\|_{\mu(\infty)},$$

where we use $p_{ij} \geq 1 \wedge c_0 \sqrt{\frac{\mu_i r}{n} \frac{\nu_j r}{n}} \log n$. We thus obtain $\|S_{ij}\|_2 \leq \frac{2}{c_0 \log n} \|Z\|_{\mu(\infty)}$ for all (i, j).

On the other hand, note that

$$\begin{aligned} \left| \mathbb{E} \left[\sum_{i,j} S_{ij}^\top S_{ij} \right] \right| &= \left| \sum_{i,j} \mathbb{E} \left[\left(\frac{1}{p_{ij}} \delta_{ij} - 1 \right)^2 \right] Z_{ij}^2 \left\| P_T(e_i e_j^\top) e_b \right\|_2^2 \cdot \frac{n}{\nu_b r} \right| \\ &= \left(\sum_{j=b,i} + \sum_{j \neq b,i} \right) \frac{1 - p_{ij}}{p_{ij}} Z_{ij}^2 \left\| P_T(e_i e_j^\top) e_b \right\|_2^2 \cdot \frac{n}{\nu_b r}. \end{aligned}$$

Applying (27), we can bound the first sum by

$$\sum_{j=b,i} \leq \sum_i \frac{1 - p_{ib}}{p_{ib}} Z_{ib}^2 \cdot 2 \left(\frac{\mu_i r}{n} + \frac{\nu_b r}{n} \right) \cdot \frac{n}{\nu_b r} \leq \frac{2}{c_0 \log n} \frac{n}{\nu_b r} \|Z_{\cdot b}\|_2^2 \leq \frac{2}{c_0 \log n} \|Z\|_{\mu(\infty,2)}^2,$$

where we use $p_{ib} \geq 1 \wedge \frac{c_0(\mu_i + \nu_b) r}{n} \log n$ in the second inequality. The second sum can be bounded using (28):

$$\begin{aligned} \sum_{j \neq b,i} &\leq \sum_{j \neq b,i} \frac{1 - p_{ij}}{p_{ij}} Z_{ij}^2 \left| e_j^\top V V^\top e_b \right|^2 \frac{n}{\nu_b r} \\ &= \frac{n}{\nu_b r} \sum_{j \neq b} \left| e_j^\top V V^\top e_b \right|^2 \sum_i \frac{1 - p_{ij}}{p_{ij}} Z_{ij}^2 \\ &\overset{(a)}{\leq} \frac{n}{\nu_b r} \sum_{j \neq b} \left| e_j^\top V V^\top e_b \right|^2 \left(\frac{1}{c_0 \log n} \sum_i Z_{ij}^2 \frac{n}{\nu_j r} \right) \\ &\leq \left(\frac{1}{c_0 \log n} \|Z\|_{\mu(\infty,2)}^2 \right) \frac{n}{\nu_b r} \sum_{j \neq b} \left| e_j^\top V V^\top e_b \right|^2 \\ &\overset{(b)}{\leq} \frac{1}{c_0 \log n} \|Z\|_{\mu(\infty,2)}^2, \end{aligned}$$

where we use $p_{ij} \geq 1 \wedge \frac{c_0 \nu_j r \log n}{n}$ in (a) and $\sum_{j \neq b} \left| e_j^\top V V^\top e_b \right|^2 \leq \left\| V V^\top e_b \right\|_2^2 \leq \frac{\nu_b r}{n}$ in (b). Combining the bounds for the two sums, we obtain $\left\| \mathbb{E} \left[\sum_{i,j} S_{ij}^\top S_{ij} \right] \right\| \leq \frac{3}{c_0 \log n} \|Z\|_{\mu(\infty,2)}^2$. We can bound $\left\| \mathbb{E} \left[\sum_{i,j} S_{ij} S_{ij}^\top \right] \right\|$ in a similar way. Applying the Matrix Bernstein inequality in Theorem 16, we have w.h.p.

$$\left\| \sqrt{\frac{n}{\nu_b r}} X_{\cdot b} \right\|_2 = \left\| \sum_{i,j} S_{ij} \right\|_2 \leq \frac{1}{2} \left(\|Z\|_{\mu(\infty)} + \|Z\|_{\mu(\infty,2)} \right)$$

for c_0 sufficiently large. Similarly we can bound $\left\| \sqrt{\frac{n}{\mu_a r}} X_{a \cdot} \right\|_2$ by the same quantity. We take a union bound over all a and b to obtain the desired results.

A.8.4 Proof of Lemma 12

Fix a matrix index (a,b) and let $w_{ab} = \sqrt{\frac{\mu_a r}{n}\frac{\nu_b r}{n}}$. We can write

$$[(P_T R_\Omega - P_T)\, Z]_{ab} \sqrt{\frac{n}{\mu_a r}}\sqrt{\frac{n}{\nu_b r}} = \sum_{i,j}\left(\frac{1}{p_{ij}}\delta_{ij} - 1\right) Z_{ij} \left\langle e_i e_j^\top, P_T(e_a e_b^\top)\right\rangle \frac{1}{w_{ab}} =: \sum_{i,j} s_{ij},$$

which is the sum of independent zero-mean variables. We first compute the following bound:

$$\begin{aligned}
&\left|\left\langle e_i e_j^\top, P_T(e_a e_b^\top)\right\rangle\right| \\
&= \left|e_i^\top U U^\top e_a e_b^\top e_j + e_i^\top (I - UU^\top) e_a e_b^\top V V^\top e_j\right| \\
&= \begin{cases}
\left|e_a^\top U U^\top e_a + e_a^\top (I-UU^\top) e_a e_b^\top V V^\top e_b\right| \le \frac{\mu_a r}{n} + \frac{\nu_b r}{n}, & i = a, j = b, \\
\left|e_a^\top (I-UU^\top) e_a e_b^\top V V^\top e_j\right| \le \left|e_b^\top V V^\top e_j\right|, & i = a, j \neq b, \\
\left|e_i^\top U U^\top e_a e_b^\top (I-VV^\top) e_b\right| \le \left|e_i^\top U U^\top e_a\right|, & i \neq a, j = b, \\
\left|e_i^\top U U^\top e_a e_b^\top V V^\top e_j\right| \le \left|e_i^\top U U^\top e_a\right| \left|e_b^\top V V^\top e_j\right|, & i \neq a, j \neq b,
\end{cases}
\end{aligned} \tag{29}$$

where we use the fact that the matrices $I - UU^\top$ and $I - VV^\top$ have spectral norm at most 1. We proceed to bound $|s_{ij}|$. Note that

$$|s_{ij}| \le (1 - \mathbb{I}(p_{ij} = 1))\, \frac{1}{p_{ij}} \cdot |Z_{ij}| \cdot \left|\left\langle e_i e_j^\top, P_T(e_a e_b^\top)\right\rangle\right| \cdot \frac{1}{w_{ab}}.$$

We distinguish four cases. When $i = a$ and $j = b$, we use (29) and $p_{ab} \ge 1 \wedge \frac{c_0(\mu_a+\nu_b) r \log^2(n)}{n}$ to obtain $|s_{ij}| \le |Z_{ij}| / (w_{ij} c_0 \log n) \le \|Z\|_{\mu(\infty)} / (c_0 \log n)$. When $i = a$ and $j \neq b$, we apply (29) to get

$$|s_{ij}| \le (1 - \mathbb{I}(p_{ij} = 1))\, \frac{|Z_{aj}|}{p_{aj}} \cdot \sqrt{\frac{\nu_b r}{n}\frac{\nu_j r}{n}} \cdot \sqrt{\frac{n}{\mu_a r}\frac{n}{\nu_b r}} \overset{(a)}{\le} |Z_{aj}| \cdot \sqrt{\frac{n}{\mu_a r}\frac{n}{\nu_j r}}\frac{1}{c_0 \log n} \le \frac{\|Z\|_{\mu(\infty)}}{c_0 \log n},$$

where (a) follows from $p_{aj} \ge \min\left\{c_0 \frac{\nu_j r \log n}{n}, 1\right\}$. In a similar fashion, we can show that the same bound holds when $i \neq a$ and $j = b$. When $i \neq a$ and $j \neq b$, we use (29) to get

$$\begin{aligned}
|s_{ij}| &\le (1 - \mathbb{I}(p_{ij} = 1))\, \frac{|Z_{ij}|}{p_{ij}} \cdot \sqrt{\frac{\mu_i r}{n}\frac{\mu_a r}{n}}\sqrt{\frac{\nu_b r}{n}\frac{\nu_j r}{n}} \cdot \sqrt{\frac{n}{\mu_a r}\frac{n}{\nu_b r}} \\
&\overset{(b)}{\le} |Z_{ij}| \cdot \sqrt{\frac{n}{\mu_i r}\frac{n}{\nu_j r}}\frac{1}{c_0 \log n} \le \frac{\|Z\|_{\mu(\infty)}}{c_0 \log n},
\end{aligned}$$

where (b) follows from $p_{ij} \ge 1 \wedge c_0 \sqrt{\frac{\mu_i r}{n}\frac{\nu_j r}{n}} \log n$ and $\max\left\{\sqrt{\frac{\mu_i r}{n}}, \sqrt{\frac{\nu_j r}{n}}\right\} \le 1$. We conclude that $|s_{ij}| \le \|Z\|_{\mu(\infty)} / (c_0 \log n)$ for all (i,j).

On the other hand, note that

$$\begin{aligned}
\left|\mathbb{E}\left[\sum_{i,j} s_{ij}^2\right]\right| &= \sum_{i,j} \mathbb{E}\left[\left(\frac{1}{p_{ij}}\delta_{ij} - 1\right)^2\right] \frac{Z_{ij}^2}{w_{ab}^2} \left\langle e_i e_j^\top, P_T(e_a e_b^\top)\right\rangle^2 \\
&= \sum_{i=a, j=b} + \sum_{i=a, j\neq b} + \sum_{i\neq a, j=b} + \sum_{i\neq a, j\neq b}.
\end{aligned}$$

We bound each of the four sums. By (29) and $p_{ab} \geq 1 \wedge \frac{c_0(\mu_a+\nu_b)r\log n}{n} \geq 1 \wedge \frac{c_0(\mu_a+\nu_b)^2 r^2 \log n}{2n^2}$, we have

$$\sum_{i=a,j=b} \leq \frac{1-p_{ab}}{p_{ab}w_{ab}^2} Z_{ab}^2 \left(\frac{\mu_a r}{n} + \frac{\nu_b r}{n}\right)^2 \leq \frac{2\left\|Z\right\|_{\mu(\infty)}^2}{c_0 \log n}.$$

By (29) and $p_{aj}w_{ab}^2 \geq w_{ab}^2 \wedge \left(c_0 w_{aj}^2 \frac{\nu_b r}{n} \log n\right)$, we have

$$\sum_{i=a,j\neq b} \leq \sum_{,j\neq b} \frac{1-p_{aj}}{p_{aj}w_{ab}^2} Z_{aj}^2 \left|e_b^\top V V^\top e_j\right| \leq \frac{\left\|Z\right\|_{\mu(\infty)}^2}{c_0 \log n} \cdot \frac{n}{\nu_b r} \sum_{j\neq b} \left|e_b^\top V V^\top e_j\right|,$$

which implies $\sum_{i=a,j\neq b} \leq \left\|Z\right\|_{\mu(\infty)}^2 / (c_0 \log n)$. Similarly we can bound $\sum_{i\neq a,j=b}$ by the same quantity. Finally, by (29) and $p_{ij} \geq 1 \wedge \left(c_0 \frac{\mu_i r}{n} \frac{\nu_j r}{n} \log n\right)$, we have

$$\begin{aligned}\sum_{i\neq a,j\neq b} &\leq \frac{1}{w_{ab}^2} \sum_{i\neq a,j\neq b} \frac{(1-p_{ij})Z_{ij}^2}{p_{ij}} \cdot \left|e_i^\top U U^\top e_a\right| \left|e_b^\top V V^\top e_j\right| \\ &\leq \frac{\left\|Z\right\|_{\mu(\infty)}^2}{c_0 \log n} \cdot \frac{1}{w_{ab}^2} \sum_{i\neq a} \left|e_i^\top U U^\top e_a\right| \sum_{j\neq b} \left|e_b^\top V V^\top e_j\right|,\end{aligned}$$

which implies $\sum_{i\neq a,j\neq b} \leq \left\|Z\right\|_{\mu(\infty)}^2 / (c_0 \log n)$. Combining pieces, we obtain

$$\left|\mathbb{E}\left[\sum_{ij} s_{ij}^2\right]\right| \leq 5\left\|Z\right\|_{\mu(\infty)}^2 / (c_0 \log n).$$

Applying the Bernstein inequality (Theorem 16), we conclude that

$$\left|\left[\left(P_T R_\Omega P_T - P_T\right) Z\right]_{ab} \sqrt{\frac{n}{\mu_a r}} \sqrt{\frac{n}{\nu_b r}}\right| = \left|\sum_{i,j} s_{ij}\right| \leq \frac{1}{2}\left\|Z\right\|_{\mu(\infty)}$$

w.h.p. for c_0 sufficiently large. The desired result follows from a union bound over all (a, b).

Appendix B. Proof of Corollary 4

Recall the setting: for each row of M, we pick it with some probability p and observe all its elements. We need a simple lemma. Let $J \subseteq [n]$ be the (random) set of the indices of the row picked, and $P_J(Z)$ be the matrix that is obtained from Z by zeroing out the rows outside J. Recall that $U\Sigma V^\top$ is the SVD of M.

Lemma 14 *If $\mu_i(M) := \frac{n}{r}\left\|U^\top e_i\right\|^2 \leq \mu_0$ for all $i \in [n]$, and $p \geq 10\frac{\mu_0 r}{n}\log\frac{2r}{\delta}$, then with probability at least $1-\delta$,*

$$\left\|U^\top P_J(U) - I_{r\times r}\right\| \leq \frac{1}{2},$$

where $I_{r\times r}$ is the identity matrix in $\mathbb{R}^{r\times r}$.

Proof Define the random variable $\eta_j := \mathbb{I}(i \in J)$ for $i = 1, 2, \ldots, n$, where $\mathbb{I}(\cdot)$ is the indicator function. Note that

$$U^\top P_J(U) - I_{r\times r} = U^\top P_J(U) - U^\top U = \sum_{i=1}^n S_{(i)} := \sum_{i=1}^n \left(\frac{1}{p}\eta_i - 1\right) U^\top e_i e_i^\top U.$$

The matrices $\{S_{(i)}\}$ are mutually independent and satisfy $\mathbb{E}\left[S_{(i)}\right] = 0$, $\left\|S_{(i)}\right\| \le \frac{1}{p}\left\|U^\top e_i\right\|_2^2 \le \frac{\mu_0 r}{pn}$, and

$$\begin{aligned}
\left\|\mathbb{E}\left[\sum_{i=1}^n S_{(i)}S_{(i)}^\top\right]\right\| = \left\|\mathbb{E}\left[\sum_{i=1}^n S_{(i)}^\top S_{(i)}\right]\right\| &= \frac{1-p}{p}\left\|\sum_{i=1}^n U^\top e_i e_i^\top U U^\top e_i e_i^\top U\right\| \\
&= \frac{1-p}{p}\left\|U^\top\left(\sum_{i=1}^n e_i e_i^\top \left\|U^\top e_i\right\|_2^2\right) U\right\| \\
&\le \frac{1}{p}\left\|\sum_{i=1}^n e_i e_i^\top \left\|U^\top e_i\right\|_2^2\right\| \\
&= \frac{1}{p}\max_i \left\|U^\top e_i\right\|_2^2 \le \frac{\mu_0 r}{pn}.
\end{aligned}$$

Note that $S_{(i)}$ are $r \times r$ matrices. It follows from the matrix Bernstein (Theorem 16) that when $p \ge \frac{10\mu_0 r}{n}\log\frac{2r}{\delta}$, we have

$$\mathbb{P}\left\{\left\|U^\top P_J(U) - I_{r\times r}\right\| \ge \frac{1}{2}\right\} \le 2r\exp\left(\frac{-(1/2)^2/2}{\frac{\mu_0 r}{6pn} + \frac{\mu_0 r}{pn}}\right) \le \delta.$$

■

Note that $\left\|U^\top P_J(U) - I_{r\times r}\right\| \le \frac{1}{2}$ implies that $U^\top P_J(U)$ is invertible, which further implies $P_J(U) \in \mathbb{R}^{n\times r}$ has rank-r. The rows picked are $P_J(M) = P_J(U)\Sigma V^\top$, which thus have full rank-r and their row space must be the same as the row space of M. Therefore, the leverage scores $\{\tilde{\nu}_j\}$ of these rows are the same as the row leverage scores $\{\nu_j(M)\}$ of M. Also note that we must have $\mu_0 \ge 1$. Setting δ and sampling Ω as described in the corollary and applying Theorem 2, we are guaranteed to recover M exactly with probability at least $1 - 9n^{-10}$. The total number of elements we have observed is

$$pn + \sum_{i,j} p_{ij} = 10\mu_0 r\log\left(\frac{2r}{4n^{-10}}\right) + c_0(\mu_0 rn + rn)\log^2 n \le c_1\mu_0 rn\log^2 n$$

for some sufficiently large universal constant c_1, and by Hoeffding's inequality, the actual number of observations is at most two times the expectation with probability at least $1 - n^{-10}$ provided c_0 is sufficiently large. The corollary follows from the union bound.

Appendix C. Proof of Theorem 6

We prove the theorem assuming $\sum_{k=1}^r \frac{1}{a_k} = \sum_{k=1}^r \frac{1}{b_k} = r$; extension to the general setting in the theorem statement will only affect the pre-constant in (4) by a factor of at most 2.

For each $k \in [r]$, let $s_k := \frac{2n}{a_k r}$, $t_k := \frac{2n}{b_k r}$. We assume the s_k's and t_k's are all integers. Under the assumption on a_k and b_k, we have $1 \le s_k, t_k \le n$ and $\sum_{k=1}^r s_k = \sum_{k=1}^r t_k = n$. Define the sets $I_k := \left\{\sum_{l=1}^{r-1} s_l + i : i \in [s_k]\right\}$ and $J_k := \left\{\sum_{l=1}^{r-1} t_l + j : j \in [t_k]\right\}$; note that $\bigcup_{k=1}^r I_k = \bigcup_{k=1}^r J_k = [n]$. The vectors $\vec{\mu}$ and $\vec{\nu}$ are given by

$$\mu_i = a_k, \quad \forall k \in [r], i \in I_k,$$
$$\nu_j = b_k, \quad \forall k \in [r], j \in J_k.$$

It is clear that $\vec{\mu}$ and $\vec{\nu}$ satisfy the property 1 in the statement of the theorem.

Let the matrix $M^{(0)}$ be given by $M^{(0)} = AB^\top$, where $A, B \in \mathbb{R}^{n \times r}$ are specified below.

- For each $k \in [r]$, we set
$$A_{ik} = \sqrt{\frac{1}{s_k}}$$
for all $i \in I_k$. All other elements of A are set to zero. Therefore, the k-th column of A has s_k non-zero elements equal to $\sqrt{\frac{1}{s_k}}$, and the columns of A have disjoint supports.
- Similarly, for each $k \in [r]$, we set
$$B_{jk} = \sqrt{\frac{1}{t_k}}$$
for all $j \in J_k$. All other elements of B are set to zero.

Observe that A is an orthonormal matrix, so

$$\mu_i\left(M^{(0)}\right) = \frac{n}{r}\left\|A_{i\cdot}\right\|_2^2 = \frac{n}{r} \cdot \frac{1}{s_k} = \frac{a_k}{2} = \frac{\mu_i}{2} \le \mu_i, \forall k \in [r], i \in I_k, .$$

A similar argument shows that $\nu_j\left(M^{(0)}\right) \le \nu_j, \forall j \in [n]$. Hence $M^{(0)} \in \mathcal{M}_r\left(\vec{\mu}, \vec{\nu}\right)$. We note that $M^{(0)}$ is a block diagonal matrix with r blocks where the k-th block has size $s_k \times t_k$, and $\left\|M^{(0)}\right\|_F = \sqrt{r}$.

Consider the i_0 and j_0 in the statement of the theorem. There must exit some $k_1, k_2 \in [r]$ such that $i_0 \in I_{k_1}$ and $j_0 \in J_{k_2}$. Assume w.l.o.g. that $s_{k_1} \ge t_{k_2}$. then

$$p_{i_0 j_0} \le \frac{\mu_{i_0} + \nu_{j_0}}{4n} \cdot r \log\left(\frac{1}{\eta}\right) = \frac{a_{k_1} + b_{k_2}}{4n} \cdot r \log\left(\frac{1}{\eta}\right) = \frac{\log\left(1/\eta\right)}{4s_{k_1}} + \frac{\log\left(1/\eta\right)}{4t_{k_2}} \le \frac{\log\left(1/\eta\right)}{2t_{k_2}},$$

where $\eta = \frac{\mu_{i_0} r}{2n} = \frac{1}{s_{k_1}}$ in part 2 of the theorem and $\eta = \frac{2}{n}$ in part 3. Because $\{p_{ij}\}$ is location-invariant w.r.t. $M^{(0)}$, we have

$$p_{ij} = p_{i_0 j_0} \le \frac{\log\left(1/\eta\right)}{2t_{k_2}}, \quad \forall i \in I_{k_1}, j \in J_{k_2}.$$

Let $W_i := |(\{i\} \times J_{k_2}) \cap \Omega|$ be the number of observed elements on $\{i\} \times J_{k_2}$. Note that for each $i \in I_{k_1}$, we have

$$\mathbb{P}\left[W_i = 0\right] = \prod_{j \in J_{k_2}} (1 - p_{ij}) \ge \left(1 - \frac{\log(1/\eta)}{2t_{k_2}}\right)^{t_{k_2}} \ge \exp\left(\log \eta\right) = \eta,$$

where we use $1-x \geq e^{-2x}, \forall 0 \leq x \leq \frac{1}{2}$ in the second inequality. Therefore, there must exist $i^* \in I_{k_1}$ for which there is no observed element in $\{i^*\} \times J_{k_2}$ with probability

$$\begin{aligned}\mathbb{P}\left[W_{i^*} = 0, \exists i^* \in I_{k_1}\right] &= 1 - \mathbb{P}\left[W_i \geq 1, \forall i \in I_{k_1}\right] \\ &\geq 1 - (1-\eta)^{s_{k_1}} \geq 1 - e^{-\eta s_{k_1}} \geq \frac{1}{2}\eta s_{k_1} \geq \begin{cases} \frac{1}{2}, & \eta = \frac{\mu_{i_0} r}{4n} \\ \frac{1}{n}, & \eta = \frac{n}{2}. \end{cases}\end{aligned}$$

These are the probabilities that appear in part 2 and part 3 of the theorem statement, respectively.

Now choose a number $\bar{s} \geq s_{k_1}$. Let $M^{(1)} = \bar{A}B^\top$, where B is the same as before and $\bar{A}$ is given by

$$\bar{A}_{ik} = \begin{cases} \sqrt{\frac{1}{\bar{s}}}, & i = i^*, k = k_2 \\ A_{ik}, & \text{otherwise.} \end{cases}$$

By varying $\bar{s}$ we can construct infinitely many such $M^{(1)}$. Clearly $M^{(1)}$ is rank-r. Observe that $M^{(1)}$ differs from $M^{(0)}$ only in the elements with indices in $\{i^*\} \times J_{k_2}$, which are not observed, so

$$M^{(0)}_{ij} = M^{(1)}_{ij}, \quad \forall (i,j) \in \Omega.$$

Also observe that any $\{p_{ij}\}$ that is location-invariant w.r.t. $M^{(0)}$ is also location-invariant w.r.t. $M^{(1)}$. The following lemma guarantees that $M^{(1)} \in \mathcal{M}_r\left(\vec{\mu}, \vec{\nu}\right)$, which completes the proof of the theorem.

Lemma 15 *The matrix $M^{(1)}$ constructed above satisfies*

$$\begin{aligned}\mu_i\left(M^{(1)}\right) &\leq 2\mu_i\left(M^{(0)}\right), \quad \forall i \in [n], \\ \nu_j\left(M^{(1)}\right) &= \nu_j\left(M^{(0)}\right), \quad \forall j \in [n].\end{aligned}$$

Proof Note that by the definition, the leverage scores of a rank-r matrix M with SVD $M = U\Sigma V^\top$ can be expressed as

$$\mu_i\left(M\right) = \frac{n}{r}\left\|U^\top e_i\right\|_2^2 = \frac{n}{r}\left\|UU^\top e_i\right\|_2^2 = \frac{n}{r}\left\|\mathcal{P}_{\mathrm{col}(M)}(e_i)\right\|_2^2,$$

where $\mathrm{col}(M)$ denotes the column space of M and $\mathcal{P}_{\mathrm{col}(M)}(\cdot)$ is the Euclidean projection onto the column space of M. A similar relation holds for the row leverage scores and the row space of M. In other words, the column/row leverage scores of a matrix are determined by its column/row space. Because $M^{(0)}$ and $M^{(1)}$ have the same row space (which is the span of the columns of B), the second set of equalities in the lemma hold.

It remains to prove the first set of inequalities for the column leverage scores. If $k_1 = k_2$, then the columns of $\bar{A}$ have unit norms and are orthogonal to each other. Using the above expression for the leverage scores, we have

$$\mu_i\left(M^{(1)}\right) = \frac{n}{r}\left\|\bar{A}\bar{A}^\top e_i\right\|_2^2 = \frac{n}{r}\left\|\bar{A}^\top e_i\right\|_2^2 = \frac{n}{r}\left\|A^\top e_i\right\|_2^2 = \mu_i\left(M^{(0)}\right).$$

If $k_1 \neq k_2$, we may assume without loss of generality that $k_1 = 1$, $k_2 = 2$ and $i^* = 1$. In the sequel we use $\bar{A}_i$ to denote the i-th columns of $\bar{A}$. We now construct two vectors $\tilde{\alpha}$ and $\tilde{\beta}$

which have the same span with $\bar{A}_1$ and $\bar{A}_2$. Define two vectors $\alpha, \beta \in \mathbb{R}^n$, such that the first s_1 elements of α and the $\{s_1+1, \ldots, s_1+s_2\}$-th elements of β are one, the first element of β is $\sqrt{\frac{s_2}{\bar{s}}}$, and all other elements of α and β are zero. Clearly $\alpha = \sqrt{s_1}\bar{A}_1$ and $\beta = \sqrt{s_2}\bar{A}_2$, so $\text{span}(\alpha, \beta) = \text{span}(\bar{A}_1, \bar{A}_2)$. We next orthogonalize α and β by letting $\bar{\alpha} = \alpha$ and

$$\bar{\beta} = \beta - \frac{\langle \alpha, \beta \rangle}{\|\alpha\|^2}\alpha = \beta - \frac{\sqrt{s_2}}{s_1\sqrt{\bar{s}}}\alpha = \begin{cases} \frac{(s_1-1)\sqrt{s_2}}{s_1\sqrt{\bar{s}}}, & i = 1 \\ -\frac{\sqrt{s_2}}{s_1\sqrt{\bar{s}}}, & i = 2, \ldots, s_1 \\ 1, & i = s_1+1, \ldots, s_1+s_2 \\ 0, & i = s_1+s_2+1, \ldots, n. \end{cases}$$

Note that $\text{span}(\bar{\alpha}, \bar{\beta}) = \text{span}(\alpha, \beta)$ and $\langle \bar{\alpha}, \bar{\beta} \rangle = 0$. Simple calculation shows that $\|\bar{\alpha}\|_2^2 = \|\alpha\|_2^2 = s_1$ and $\|\bar{\beta}\|_2^2 = \left(\frac{s_1-1}{s_1\bar{s}} + 1\right) s_2$. Finally, we normalize $\bar{\alpha}$ and $\bar{\beta}$ by letting $\tilde{\alpha} = \bar{\alpha}/\|\bar{\alpha}\|$ and $\tilde{\beta} = \bar{\beta}/\|\bar{\beta}\|$. It is clear that $\text{span}(\tilde{\alpha}, \tilde{\beta}) = \text{span}(\bar{A}_1, \bar{A}_2)$, and $\langle \tilde{\alpha}, \bar{A}_k \rangle = \left\langle \tilde{\beta}, \bar{A}_k \right\rangle = 0, \forall k = 3, \ldots, r$.

Now consider the matrix $\tilde{A} \in \mathbb{R}^{n \times r}$ obtained from $\bar{A}$ by replacing the first two columns of $\bar{A}$ with $\tilde{\alpha}$ and $\tilde{\beta}$, respectively. Because $\text{col}(\tilde{A}) = \text{col}(\bar{A}) = \text{col}(M^{(1)})$, we have

$$\mu_i\left(M^{(1)}\right) = \frac{n}{r}\left\|\mathcal{P}_{\text{col}(\tilde{A})}\left(e_i\right)\right\|^2.$$

But the columns of $\tilde{A}$ have unit norms and are orthogonal to each other. It follows that

$$\mu_i\left(M^{(1)}\right) = \frac{n}{r}\left\|\tilde{A}\tilde{A}^\top e_i\right\|^2 = \frac{n}{r}\left\|\tilde{A}^\top e_i\right\|^2.$$

For $s_1+s_2 < i \le n$, since $\bar{s} \ge s_1$ we have $\left\|\tilde{A}^\top e_i\right\|^2 = \left\|\bar{A}^\top e_i\right\|^2 = \left\|A^\top e_i\right\|^2$ so $\mu_i\left(M^{(1)}\right) = \mu_i\left(M^{(0)}\right)$. For $i \in [s_1+s_2]$, we have

$$\left\|\tilde{A}^\top e_i\right\|^2 = \tilde{\alpha}_i^2 + \tilde{\beta}_i^2 = \begin{cases} \frac{1}{s_1} + \frac{(s_1-1)^2}{s_1(s_1-1)+s_1^2\bar{s}} \le \frac{2}{s_1} = 2\left\|A^\top e_i\right\|^2, & i = 1 \\ \frac{1}{s_1} + \frac{1}{s_1(s_1-1)+s_1^2\bar{s}} \le \frac{2}{s_1} = 2\left\|A^\top e_i\right\|^2, & i = 2, \ldots, s_1 \\ \frac{s_1\bar{s}}{(s_1-1+s_1\bar{s})s_2} \le \frac{1}{s_2} = \left\|A^\top e_i\right\|^2, & i = s_1+1, \ldots, s_1+s_2. \end{cases}$$

This means

$$\mu_i\left(M^{(1)}\right) \le \frac{2n}{r}\left\|A^\top e_i\right\|^2 = 2\mu_i(M^{(0)}), \forall i \in [s_1+s_2],$$

which completes the proof of the lemma. ■

Appendix D. Proof of Theorem 7

Suppose the rank-r SVD of $\bar{M}$ is $\bar{U}\bar{\Sigma}\bar{V}^\top$; so $\bar{U}\bar{\Sigma}\bar{V}^\top = RMC = RU\Sigma V^\top C$. By definition, we have

$$\frac{\bar{\mu}_i r}{n} = \left\|P_{\tilde{U}}(e_i)\right\|_2^2,$$

where $P_{\tilde{U}}(\cdot)$ denotes the projection onto the column space of $\tilde{U}$, which is the same as the column space of RU. This projection has the explicit form

$$P_{\tilde{U}}(e_i) = RU\left(U^\top R^2 U\right)^{-1} U^\top R e_i.$$

It follows that

$$\begin{aligned}\frac{\bar{\mu}_i r}{n} &= \left\| RU\left(U^\top R^2 U\right)^{-1} U^\top R e_i \right\|_2^2 \\ &= R_i^2 e_i^\top U\left(U^\top R^2 U\right)^{-1} U^\top e_i \\ &\le R_i^2 \left[\sigma_r\left(RU\right)\right]^{-2} \left\| U^\top e_i \right\|_2^2 \\ &\le R_i^2 \frac{\mu_0 r}{n} \left[\sigma_r\left(RU\right)\right]^{-2}, \end{aligned} \tag{30}$$

where $\sigma_r(\cdot)$ denotes the r-th singular value and the last inequality follows from the standard incoherence assumption $\max_{i,j}\{\mu_i, \nu_j\} \le \mu_0$. We now bound $\sigma_r\left(RU\right)$. Since RU has rank r, we have

$$\sigma_r^2\left(RU\right) = \min_{\|x\|=1} \|RUx\|_2^2 = \min_{\|x\|=1} \sum_{i=1}^n R_i^2 \left| e_i^\top U x \right|^2. \tag{31}$$

If we let $z_i := \left|e_i^\top U x\right|^2$ for each $i \in [n]$, then z_i satisfies

$$\sum_{i=1}^n z_i = \|Ux\|_2^2 = \|x\|_2^2 = 1$$

and by the standard incoherence assumption,

$$z_i \le \left\| U^\top e_i \right\|_2^2 \|x\|_2^2 \le \frac{\mu_0 r}{n}.$$

Therefore, the value of the minimization (31) is lower-bounded by the optimal value of the following program

$$\begin{aligned} \min_{z \in \mathbb{R}^n} \ & \sum_{i=1}^n R_i^2 z_i \\ \text{s.t.} \ & \sum_{i=1}^n z_i = 1, \quad 0 \le z_i \le \frac{\mu_0 r}{n},\ i = 1, \dots, n. \end{aligned} \tag{32}$$

From the theory of linear programming, we know the minimum is achieved at an extreme point z^* of the feasible set. Such an extreme point z^* satisfies $z_i^* \ge 0, \forall i$ and n linear equalities

$$\begin{aligned} \sum_{i=1}^n z_i^* &= 1, \\ z_i^* &= 0, \quad \text{for } i \in I_1, \\ z_i^* &= \frac{\mu_0 r}{n}, \text{ for } i \in I_2 \end{aligned}$$

for some index sets I_1 and I_2 such that $I_1 \cap I_2 = \phi$, $|I_1| + |I_2| = n - 1$. It is easy to see that we must have $|I_2| = \left\lfloor \frac{n}{\mu_0 r} \right\rfloor$. Since $R_1 \le R_2 \le \ldots \le R_n$, the minimizer z^* has the form

$$z_i^* = \begin{cases} \frac{\mu_0 r}{n}, & i = 1, \ldots, \left\lfloor \frac{n}{\mu_0 r} \right\rfloor, \\ 1 - \left\lfloor \frac{n}{\mu_0 r} \right\rfloor \cdot \frac{\mu_0 r}{n}, & i = \left\lfloor \frac{n}{\mu_0 r} \right\rfloor + 1, \\ 0, & i = \left\lfloor \frac{n}{\mu_0 r} \right\rfloor + 2, \ldots, n, \end{cases}$$

and the value of the minimization (32) is at least

$$\sum_{i=1}^{\lfloor n/(\mu_0 r) \rfloor} R_i^2 \frac{\mu_0 r}{n}.$$

This proves that $\sigma_r^2(RU) \ge \frac{\mu_0 r}{n} \sum_{i=1}^{\lfloor n/(\mu_0 r) \rfloor} R_i^2$. Combining with (30), we obtain that

$$\frac{\bar{\mu}_i r}{n} \le \frac{R_i^2}{\sum_{i'=1}^{\lfloor n/(\mu_0 r) \rfloor} R_i^2}, \quad \frac{\bar{\nu}_j r}{n} \le \frac{C_j^2}{\sum_{j'=1}^{\lfloor n/(\mu_0 r) \rfloor} C_{j'}^2};$$

the proof for $\bar{\nu}_j$ is similar. Applying Theorem 2 to the equivalent problem (9) with the above bounds on $\bar{\mu}_i$ and $\bar{\nu}_j$ proves the theorem.

Appendix E. Weighted vs Unweighted Nuclear Norm Minimization for Non-uniform Sampling

In this section we provide a concrete example of the gain of weighting under the setting of Section 5.1, where the observed entries are given and distributed non-uniformly. Suppose M is an n-by-n matrix with rank r, and its incoherence parameter satisfies $\mu_0 r = c$, where c is a numerical constant. We assume the sampling probabilities have the form $p_i^{\mathrm{r}} = p_i^{\mathrm{c}} = \min\{\gamma \frac{i^{0.15} \log n}{n^{0.65}}, 1\}$ for $i = 1, 2, \ldots, n$; here the minimization ensures $p_i^{\mathrm{r}} p_j^{\mathrm{c}}$ is a probability. Note that the parameter γ determines the expected number of samples $\sum_{i,j} p_i^{\mathrm{r}} p_j^{\mathrm{c}}$. For the condition (11) for the unweighted approach to hold, we need $\gamma^2 \gtrsim n^{0.3}$, and thus the the expected number of samples is at least

$$\sum_{i,j} p_i^{\mathrm{r}} p_j^{\mathrm{c}} \ge \sum_{i,j} \gamma \frac{i^{0.15}}{n^{0.65}} \cdot \gamma \frac{j^{0.15}}{n^{0.65}} = \Omega(n^{1.3}),$$

where we use the estimate $\sum_{i=1}^n i^{0.15} = \Theta(n^{1.15})$. On the other hand, the condition (10) for the weighted approach is satisfied as long as $\gamma^2 \gtrsim n^{0.15}$, so the the expected number of samples satisfies

$$\sum_{i,j} p_i^{\mathrm{r}} p_j^{\mathrm{c}} \le \sum_{i,j} \gamma \frac{i^{0.15}}{n^{0.65}} \cdot \gamma \frac{j^{0.15}}{n^{0.65}} \cdot \log^2 n = O(n^{1.15} \log^2 n)$$

when $\gamma^2 = \Theta(n^{0.15})$. Therefore, the number of samples required by the condition (10) for the weighted approach is *order-wise* smaller than the unweighted counterpart (11). Note that the conditions (10) and (11) are the *best known sufficient* conditions for exact matrix completion using the weighted and unweighted approaches, respectively, so the comparison above suggests a significant gain in sample complexity using the weighted approach.

Appendix F. Matrix Bernstein Inequality

Theorem 16 (Tropp 2012) *Let $X_1, \ldots, X_N \in \mathbb{R}^{n_1 \times n_2}$ be independent zero mean random matrices. Suppose*

$$\max\left\{\left\|\mathbb{E}\sum_{k=1}^{N} X_k X_k^\top\right\|, \left\|\mathbb{E}\sum_{k=1}^{N} X_k^\top X_k\right\|\right\} \leq \sigma^2$$

and $\|X_k\| \leq B$ almost surely for all k. Then we have

$$\mathbb{P}\left\{\left\|\sum_{k=1}^{N} X_k\right\| \geq t\right\} \leq (n_1+n_2)\exp\left(\frac{-t^2/2}{Bt/3+\sigma^2}\right)$$

As a consequence, for any $c > 0$, we have

$$\left\|\sum_{k=1}^{N} X_k\right\| \leq 2\sqrt{c\sigma^2\log(n_1+n_2)} + cB\log(n_1+n_2). \tag{33}$$

with probability at least $1-(n_1+n_2)^{-(c-1)}$.

References

D. Achlioptas and F. McSherry. Fast computation of low-rank matrix approximations. *Journal of the ACM*, 54(2):9, 2007.

D. Achlioptas, Z. Karnin, and E. Liberty. Matrix entry-wise sampling: simple is best. *http://cs-www.cs.yale.edu/homes/el327/papers/matrixSampling.pdf*, 2013.

S. Arora, E. Hazan, and S. Kale. A fast random sampling algorithm for sparsifying matrices. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 272–279. Springer, 2006.

C. Boutsidis, M. Mahoney, and P. Drineas. An improved approximation algorithm for the column subset selection problem. In *Proceedings of the Symposium on Discrete Algorithms*, pages 968–977, 2009.

N. Burq, S. Dyatlov, R. Ward, and M. Zworski. Weighted eigenfunction estimates with applications to compressed sensing. *SIAM Journal on Mathematical Analysis*, 44(5): 3481–3501, 2012.

J. Cai, E. Candès, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.

E. Candès and Y. Plan. Matrix completion with noise. *Proceedings of the IEEE*, 98(6): 925–936, 2010.

E. Candès and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717–772, 2009.

E. Candès and T. Tao. The power of convex relaxation: Near-optimal matrix completion. *IEEE Transactions on Information Theory*, 56(5):2053–2080, 2010.

E. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of the ACM*, 58(3):11, 2011.

V. Chandrasekaran, S. Sanghavi, P. Parrilo, and A. Willsky. Rank-sparsity incoherence for matrix decomposition. *SIAM Journal on Optimization*, 21(2):572–596, 2011.

S. Chatterjee and A. Hadi. Influential observations, high leverage points, and outliers in linear regression. *Statistical Science*, 1(3):379–393, 1986.

Y. Chen. Incoherence-optimal matrix completion. *IEEE Transactions on Information Theory*, 61(5):2909–2923, 2015.

Y. Chen, A. Jalali, S. Sanghavi, and C. Caramanis. Low-rank matrix recovery from errors and erasures. *IEEE Transactions on Information Theory*, 59(7):4324–4337, 2013.

P. Drineas and A. Zouzias. A note on element-wise matrix sparsification via a matrix-valued Bernstein inequality. *Information Processing Letters*, 111(8):385–389, 2011.

P. Drineas, M. Magdon-Ismail, M. Mahoney, and D. Woodruff. Fast approximation of matrix coherence and statistical leverage. *Journal of Machine Learning Research*, 13: 3475–3506, 2012.

M. Fazel. *Matrix Rank Minimization with Applications*. PhD thesis, Stanford University, 2002.

R. Foygel, O. Shamir, N. Srebro, and R. Salakhutdinov. Learning with the weighted trace-norm under arbitrary sampling distributions. In *Advances in Neural Information Processing Systems 24*, pages 2133–2141. 2011.

D. Gross. Recovering low-rank matrices from few coefficients in any basis. *IEEE Transactions on Information Theory*, 57(3):1548–1566, 2011.

P. Jain, P. Netrapalli, and S. Sanghavi. Low-rank matrix completion using alternating minimization. In *Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing*, pages 665–674. ACM, 2013.

R. H. Keshavan, A. Montanari, and S. Oh. Matrix completion from a few entries. *IEEE Transactions on Information Theory*, 56(6):2980–2998, 2010.

F. Krahmer and R. Ward. Stable and robust sampling strategies for compressive imaging. *IEEE Transactions on Image Processing*, 23(2):612–622, 2014.

A. Krishnamurthy and A. Singh. Low-rank matrix and tensor completion via adaptive sampling. In *Advances in Neural Information Processing Systems 26*, pages 836–844, 2013.

A. Krishnamurthy and A. Singh. On the power of adaptivity in matrix completion and approximation. *arXiv preprint arXiv:1407.3619*, 2014.

Z. Lin, M. Chen, L. Wu, and Y. Ma. The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices. *UIUC Technical Report UILU-ENG-09-2215*, 2009.

M. Mahoney. Randomized algorithms for matrices and data. *Foundations and Trends in Machine Learning*, 3(2):123–224, 2011.

S. Negahban and M. Wainwright. Restricted strong convexity and weighted matrix completion: Optimal bounds with noise. *Journal of Machine Learning Research*, 13:1665–1697, 2012.

H. Rauhut and R. Ward. Sparse Legendre expansions via ℓ_1-minimization. *Journal of Approximation Theory*, 164(5):517–533, 2012.

B. Recht. A simpler approach to matrix completion. *Journal of Machine Learning Research*, 12:3413–3430, 2011.

D. Spielman and N. Srivastava. Graph sparsification by effective resistances. *SIAM Journal on Computing*, 40(6):1913–1926, 2011.

N. Srebro and R. Salakhutdinov. Collaborative filtering in a non-uniform world: Learning with the weighted trace norm. In *Advances in Neural Information Processing Systems*, pages 2056–2064, 2010.

J. Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of Computational Mathematics*, 12(4):389–434, 2012.

X. Yang and G. Karniadakis. Reweighted ℓ_1 minimization method for stochastic elliptic differential equations. *Journal of Computational Physics*, 248:87–108, 2013.

Journal of Machine Learning Research 16 (2015) 3035-3078 Submitted 8/14; Revised 3/15; Published 12/15

Eigenwords: Spectral Word Embeddings

Paramveer S. Dhillon* DHILLON@MIT.EDU
Sloan School of Management
Massachusetts Institute of Technology
Cambridge, MA 02142, USA

Dean P. Foster FOSTER@WHARTON.UPENN.EDU
Department of Statistics
The Wharton School, University of Pennsylvania
Philadelphia, PA 19104, USA

Lyle H. Ungar UNGAR@CIS.UPENN.EDU
Department of Computer and Information Science
University of Pennsylvania
Philadelphia, PA 19104, USA

Editor: Ivan Titov

Abstract

Spectral learning algorithms have recently become popular in data-rich domains, driven in part by recent advances in large scale randomized SVD, and in spectral estimation of Hidden Markov Models. Extensions of these methods lead to statistical estimation algorithms which are not only fast, scalable, and useful on real data sets, but are also provably correct. Following this line of research, we propose four fast and scalable spectral algorithms for learning word embeddings – low dimensional real vectors (called *Eigenwords*) that capture the "meaning" of words from their context. All the proposed algorithms harness the multi-view nature of text data i.e. the left and right context of each word, are fast to train and have strong theoretical properties. Some of the variants also have lower sample complexity and hence higher statistical power for rare words. We provide theory which establishes relationships between these algorithms and optimality criteria for the estimates they provide. We also perform thorough qualitative and quantitative evaluation of *Eigenwords* showing that simple linear approaches give performance comparable to or superior than the state-of-the-art non-linear deep learning based methods.

Keywords: spectral learning, CCA, word embeddings, NLP

1. Introduction

In recent years there has been immense interest in learning embeddings for words from large amounts of raw text[1]. Word embeddings map each word in text to a 'k' dimensional (~ 50) real valued vector. They are typically learned in a totally unsupervised manner by exploiting the co-occurrence structure of words in unlabeled text. Ideally these embeddings should capture a rich variety of information about that word, including topic, part of speech,

∗. This work was done when PSD was a graduate student at the University of Pennsylvania.
1. This paper is based in part on work in (Dhillon et al., 2011),(Dhillon et al., 2012b).

word features such as animacy, sentiment, gender, whether the numbers are years or small numbers, and the direction of sentiment (happy vs. sad).

The importance of word embeddings has been amplified by the fact that over the past decade there has been increased interest in using unlabeled data to supplement the labeled data in semi-supervised learning. Semi-supervised learning reduces data sparsity and gives improved generalization accuracies in high dimensional domains like NLP. Approaches like (Ando and Zhang, 2005; Suzuki and Isozaki, 2008) have been empirically very successful, achieving excellent accuracies on a variety of NLP tasks. However, it is often difficult to adapt these approaches to use in conjunction with an existing supervised NLP system as they enforce a particular choice of model.

An increasingly popular alternative is to learn representational embeddings for words from a large collection of unlabeled data, either using a generative model or an artificial neural network, and to use these embeddings to augment the feature set of a supervised learner, thereby improving the performance of a state-of-the-art NLP system such as a sentiment analyzer, parser or part of speech tagger.

Word embeddings have proven useful and have given state-of-the-art performance on many natural language processing tasks e.g. syntactic parsing (Täckström et al., 2012; Parikh et al., 2014), POS Tagging (Dhillon et al., 2012b; Huang et al., 2013), dependency parsing (Bansal et al., 2014; Koo et al., 2008; Dhillon et al., 2012a), sentiment analysis (Dhillon et al., 2012b), chunking (Turian et al., 2010; Dhillon et al., 2011), Named Entity Recognition (NER) (Turian et al., 2010; Dhillon et al., 2011), word analogies (Mikolov et al., 2013a,b) and word similarity (Huang et al., 2012) to name a few.

These NLP systems use labeled data to learn a model, but there is often only a limited amount of labeled text available for these tasks. (This is less of a problem for English, but other languages often have very little labeled data.) Thus, word embeddings, which can be learned from large amounts of unlabeled data, provide a highly discriminative set of features which enable the supervised learner to perform better.

As mentioned earlier, embedding methods produce features in low dimensional spaces, unlike the traditional approach of working in the original high dimensional vocabulary space with only one dimension "on" at a given time.

Broadly speaking, embedding methods fall into two categories:

1. *Clustering based word embeddings*: Clustering methods, often hierarchical, are used to group distributionally similar words based on their contexts. The two dominant approaches are Brown Clustering (Brown et al., 1992) and (Pereira et al., 1993). As recently shown, HMMs can also be used to induce a multinomial distribution over possible clusters (Huang and Yates, 2009).

2. *Dense embeddings*: These embeddings are dense, low dimensional and real-valued. Each dimension of these embeddings captures latent information about a combination of syntactic and semantic word properties. They can either be induced using neural networks like C&W embeddings (Collobert and Weston, 2008), *Hierarchical log-linear* (HLBL) embeddings (Mnih and Hinton, 2007), `word2vec` embeddings (Mikolov et al., 2013a,b) or by eigen-decomposition of the word co-occurrence matrix, e.g. *Latent Semantic Analysis/Latent Semantic Indexing* (LSA/LSI) (Dumais et al., 1988).

The most classic and successful algorithm for learning word embeddings is Latent Semantic Analysis (LSA) (Landauer et al., 2008), which works by performing SVD on the word by document matrix.

Unfortunately, the state-of-the-art embedding methods suffer from a number of shortcomings: 1). They are slow to train (especially, the Deep Learning based approaches (Collobert and Weston, 2008; Mnih and Hinton, 2007). Recently, (Mikolov et al., 2013a,b) have proposed neural network based embeddings which avoid using the hidden layers which are typical in Deep Learning. This, coupled with good engineering allows their embeddings to be trained in minutes. 2). Are sensitive to the scaling of the embeddings (especially ℓ_2 based approaches like LSA/PCA). 3). Learn a single embedding for a given word type; i.e. all the occurrences of the word *"bank"* will have the same embedding, irrespective of whether the context of the word suggests it means *"a financial institution"* or *"a river bank."* Recently, (Huang et al., 2012) have proposed context specific word embeddings, but their Deep Learning based approach is slow and can not scale to large vocabularies.

In this paper we provide spectral algorithms (based on eigen-decomposition) for learning word embeddings, as they have been shown to be fast and scalable for learning from large amounts of unlabeled data (Turney and Pantel, 2010), have a strong theoretical grounding, and are guaranteed to converge to globally optimal solutions (Hsu et al., 2009). Particularly, we are interested in Canonical Correlation Analysis (CCA) (Hotelling, 1935) based methods since:

1. Unlike PCA or LSA based methods, they are scale invariant and
2. Unlike LSA, they can capture multi-view information. In text applications the left and right contexts of the words provide a natural split into two views which is totally ignored by LSA as it throws the entire context into a bag of words while constructing the term-document matrix.

We propose a variety of dense embeddings; they learn real-valued word embeddings by performing Canonical Correlation Analysis (CCA) (Hotelling, 1935) between the past and future views of the data. All our embeddings have a number of common characteristics and address the shortcomings of the current state-of-the-art embeddings. In particular, they are:

1. Fast, scalable and scale invariant.
2. Provide better sample complexity[2] for rare words.
3. Can induce context-specific embeddings i.e. different embeddings for *"bank"* based on whether it means *"a financial institution"* or *"a river bank."*
4. Have strong theoretical foundations.

Most importantly, in this paper we show that simple linear methods based on eigen-decomposition of the context matrices at the simplest level give accuracies comparable to or better than state-of-the-art highly non-linear deep learning based approaches like (Collobert and Weston, 2008; Mnih and Hinton, 2007; Mikolov et al., 2013a,b).

2. In the sense that relative statistical efficiency is better.

The remainder of the paper is organized as follows. In the next section we give a brief overview of CCA, which forms the core of our method. The following section describes our four proposed algorithms. After a brief description of context-specific embeddings and of the efficient SVD method we use, we present a set of systematic studies. These studies evaluate our CCA variants and alternatives including those derived from deep neural networks, including C&W, HLB, SENNA, and word2vec on problems in POS tagging, word similarity, generalized sentiment classification, NER, cross-lingual WSD and semantic & syntactic analogies.

2. Brief Review: Canonical Correlation Analysis (CCA)

CCA (Hotelling, 1935) is the analog to Principal Component Analysis (PCA) for pairs of matrices. PCA computes the directions of maximum covariance between elements in a single matrix, whereas CCA computes the directions of maximal correlation between a pair of matrices. Like PCA, CCA can be cast as an eigenvalue problem on a covariance matrix, but can also be interpreted as deriving from a generative mixture model (Bach and Jordan, 2005). See (Hardoon et al., 2004) for a review of CCA with applications to machine learning.

More specifically, given n i.i.d samples from two sets of multivariate data $\mathcal{D}_z = \{z_1, \ldots, z_n\} \in \mathbb{R}^{m_1}$ and $\mathcal{D}_w = \{w_1, \ldots, w_n\} \in \mathbb{R}^{m_2}$ where pairs (z_1, w_1) have correspondence and so on, CCA tries to find a pair of linear transformations $\boldsymbol{\phi}_z \in \mathbb{R}^{m_1 \times k}$ and $\boldsymbol{\phi}_w \in \mathbb{R}^{m_2 \times k}$, (where $k \leq m_1 \leq m_2$) such that the correlation between the projection of z onto $\boldsymbol{\phi}_z$ and w onto $\boldsymbol{\phi}_w$ is maximized. This can be expressed as the following optimization problem:

$$\max_{\boldsymbol{\phi}_z, \boldsymbol{\phi}_w} \frac{\boldsymbol{\phi}_z^\top \mathbf{C}_{zw} \boldsymbol{\phi}_w}{\sqrt{\boldsymbol{\phi}_z^\top \mathbf{C}_{zz} \boldsymbol{\phi}_z} \sqrt{\boldsymbol{\phi}_w^\top \mathbf{C}_{ww} \boldsymbol{\phi}_w}},$$

where $\mathbf{C}_{zw}$ $(=\sum_{i=1}^n (z_i - \mu_z)^\top (w_i - \mu_w))$, $\mathbf{C}_{ww}$ $(=\sum_{i=1}^n (w_i - \mu_w)^\top (w_i - \mu_w))$, and $\mathbf{C}_{zz}$ $(=\sum_{i=1}^n (z_i - \mu_z)^\top (z_i - \mu_z))$, are the sample covariance matrices and $\mu_{(\cdot)}$ are the sample means.

The above optimization problem can be solved via simple eigendecomposition (e.g. using $eig()$ function in MATLAB or R). The left and right canonical correlates (ϕ_z, ϕ_w) are the 'k' principal eigenvectors corresponding to the $\lambda_1 \geq, \ldots, \geq \lambda_k$ eigenvalues of the following equations:

$$\mathbf{C}_{zz}^{-1} \mathbf{C}_{zw} \mathbf{C}_{ww}^{-1} \mathbf{C}_{wz} \boldsymbol{\phi}_z = \lambda \boldsymbol{\phi}_z,$$
$$\mathbf{C}_{ww}^{-1} \mathbf{C}_{wz} \mathbf{C}_{zz}^{-1} \mathbf{C}_{zw} \boldsymbol{\phi}_w = \lambda \boldsymbol{\phi}_w.$$

There is an equivalent formulation of CCA which allows us to compute the solution via SVD of ${\mathbf{C}_{zz}}^{-1/2} \mathbf{C}_{zw} {\mathbf{C}_{ww}}^{-1/2}$. (See the appendix for proof.)

$$ {\mathbf{C}_{zz}}^{-1/2} \mathbf{C}_{zw} {\mathbf{C}_{ww}}^{-1/2} = \boldsymbol{\phi}_z \Lambda \boldsymbol{\phi}_w^\top, \tag{1}$$

where $(\boldsymbol{\phi}_z, \boldsymbol{\phi}_w)$ are the left and right singular vectors and Λ is the diagonal matrix of singular values. Finally, the CCA projections are gotten by "de-whitening"[3] as ${\boldsymbol{\phi}_z}^{proj} = \mathbf{C}_{zz}^{-1/2} \boldsymbol{\phi}_z$ and ${\boldsymbol{\phi}_w}^{proj} = \mathbf{C}_{ww}^{-1/2} \boldsymbol{\phi}_w$.

3. One way to think about CCA is as "whitening" the covariance matrix. Whitening is a decorrelation transformation that transforms a set of random variables with an arbitrary covariance matrix into a set

For most of the embeddings proposed in this paper, the SVD formulation (Equation 1) is preferred since it requires fewer multiplications of large sparse matrices which is an expensive operation. Hence, we define the operation $(\boldsymbol{\phi}_z{}^{proj}, \boldsymbol{\phi}_w{}^{proj}) \equiv \text{CCA}(\mathbf{Z}, \mathbf{W})$, where $\mathbf{Z}$ ($\in \mathbb{R}^{n \times m_1}$) and $\mathbf{W}$ ($\in \mathbb{R}^{n \times m_2}$) are the matrices constructed from the data $\mathcal{D}_z$ and $\mathcal{D}_w$ respectively.

2.1 Suitability of CCA for Learning Word Embeddings

Recently, (Foster et al., 2008) showed that CCA can exploit multi-view nature of the data and provide sufficient conditions for CCA to achieve dimensionality reduction without losing predictive power. They assume that the data was generated by the model shown in Figure 1. The two assumptions that they make are that 1) Each of the two views are independent conditional on a k-dimensional hidden state $\hbar$ and that 2) The two views provide a redundant estimate of the hidden state $\hbar$.

These two assumptions are generalization of the assumptions made by co-training (Blum and Mitchell, 1998) (Figure 2), as co-training conditions on the observed labels y and not on a more flexible representation i.e. a hidden state $\hbar$.

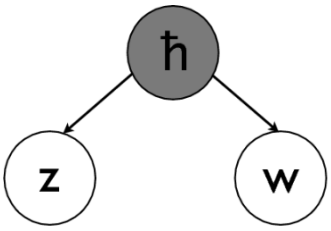


Figure 1: Multi-View Assumption. Grey color indicates that the state is hidden.

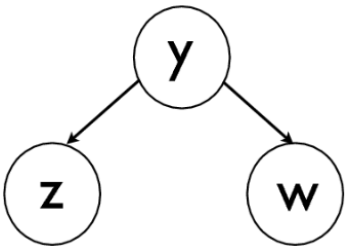


Figure 2: Co-training Assumption.

In text and Natural Language Processing (NLP) applications, its typical to assume a Hidden Markov Model (HMM) as the data generating model (Jurafsky and Martin, 2000). Its easy to see that a Hidden Markov Model (HMM) satisfies the multi-view assumption. Hence, the left and right context of a given word provides two natural views and one could use CCA to estimate the hidden state $\hbar$.

of new random variables whose covariance is the identity matrix i.e. they are uncorrelated. De-whitening, on the other hand, transforms the set of random variables to have a covariance matrix that is not an identity matrix.

Furthermore, as mentioned earlier, CCA is scale invariant and provides a natural scaling (inverse or square root of the inverse of the auto-covariance matrix, depending on whether we use Eigen-decomposition or SVD formation) for the observations. If we further use the SVD formulation, then it also allows us to harness the recent advances in large scale randomized SVD (Halko et al., 2011), which allows the embeddings learning algorithms to be fast and scalable.

The invariance of CCA to linear data transformations allows proofs that keeping the dominant singular vectors (those with largest singular values) will faithfully capture any state information (Kakade and Foster, 2007). Also, CCA extends more naturally than LSA to sequences of words[4]. Remember that LSA uses "bags of words," which are good for capturing topic information, but fail for problems like part of speech (POS) tagging which need sequence information.

Finally, as we show in the next section the CCA formulation can be naturally extended to a two step procedure that, while equivalent in the limit of infinite data, gives higher accuracies for finite corpora and provides better sample complexity.

So, in summary we estimate a hidden state associated with words by computing the dominant canonical correlations between target words and the words in their immediate context. The main computation, finding the singular value decomposition of a scaled version of the co-occurrence matrix of counts of words with their contexts, can be done highly efficiently. Use of CCA also allows us to prove theorems about the optimality of our reconstruction of the state.

In the next section we show how to efficiently compute a vector that characterizes each word type by using the left singular values of the above CCA to map from the word space (size v) to the state space (size k). We call this mapping the *eigenword dictionary* for words, as it associates with every word a vector that captures that word's syntactic and semantic attributes. As will be made clear below, the *eigenword dictionary* is arbitrary up to a rotation, but captures the information needed for any linear model to predict properties of the words such as part of speech or word sense.

3. Problem Formulation

Our goal is to estimate a vector for each word *type* that captures the distributional properties of that word in the form of a low dimensional representation of the correlation between that word and the words in its immediate context.

More formally, assume a document (in practice a concatenation of a large number of documents) consisting of n tokens $\{w_1, w_2, ..., w_n\}$, each drawn from a vocabulary of v words. Define the left and right contexts of each token w_i as the h words to the left or right of that token. The context sits in a very high dimensional space, since for a vocabulary of size v, each of the $2h$ words in the combined context requires an indicator function of dimension v. The tokens themselves sit in a v dimensional space of words which we want to project down to a k dimensional state space. We call the mapping from word types to their latent vectors the *eigenword dictionary*.

4. It is important to note that it is possible to come up with PCA variants which take sequence information into account.

For a set of documents containing n tokens, define $\mathbf{L}, \mathbf{R} \in \mathbb{R}^{n \times vh}$ as the matrices specifying the left and right contexts of the tokens, and $\mathbf{W} \in \mathbb{R}^{n \times v}$ as the matrix of the tokens themselves. In $\mathbf{W}$, we represent the presence of the j^{th} word type in the i^{th} position in a document by setting matrix element $w_{ij} = 1$. $\mathbf{L}$ and $\mathbf{R}$ are similar, but have columns for each word in each position in the context. (For example, in the sentence "I ate green apples yesterday.", for a context of size $h = 2$, the left context of "green" would be "I ate" and the right context would be "apples yesterday" and the third row of $\mathbf{W}$ would have a "1" in the column corresponding to the word "green.")

Define the complete context matrix $\mathbf{C}$ as the concatenation [$\mathbf{L}$ $\mathbf{R}$]. Thus, for a trigram representation with vocabulary size v words, history size $h = 1$, $\mathbf{C}$ has $2v$ columns – one for each possible word to the left of the target word and one for each possible word to the right of the target word.

$\mathbf{W}^\top\mathbf{C}$ then contains the counts of how often each word w occurs in each context c, the matrix $\mathbf{C}^\top\mathbf{C}$ gives the covariance of the contexts, and $\mathbf{W}^\top\mathbf{W}$, the word covariance matrix, is a diagonal matrix with the counts of each word on the diagonal[5].

All the matrices i.e. $\mathbf{L}$, $\mathbf{R}$, $\mathbf{W}$ and $\mathbf{C}$, are instantiations of the underlying multivariate random variables l, r, w and c of dimensions vh, vh, v and $2vh$ respectively. We define these multivariate random variables as we will operate on them to prove the theoretical properties of some of our algorithms.

We want to find a vector representation of each of the v word types such that words that are distributionally similar (ones that have similar contexts) have similar state vectors. We will do this using Canonical Correlation Analysis (CCA) (Hotelling, 1935; Hardoon and Shawe-Taylor, 2008), by taking the CCA between the combined left and right contexts $\mathbf{C} = [\mathbf{L}\ \mathbf{R}]$ and their associated tokens, $\mathbf{W}$.

3.1 One Step CCA (OSCCA)

Using the above, we can define a "One step CCA" (OSCCA), procedure to estimate the *eigenword dictionary* as follows:

$$(\boldsymbol{\phi}_w, \boldsymbol{\phi}_c) = CCA(\mathbf{W}, \mathbf{C}), \tag{2}$$

where the $v \times k$ matrix $\boldsymbol{\phi}_w$ contains the *eigenword dictionary* that characterizes each of the v words in the vocabulary using a k dimensional vector. More generally, the "state" vectors $\mathbf{S}$ for the n tokens can be estimated either from the context as $\mathbf{C}\phi_c$ or (trivially) from the tokens themselves as $\mathbf{W}\boldsymbol{\phi}_w$. Its important to note that both these estimation procedures give a redundant estimate of the same hidden "state."

The left canonical correlates found by OSCCA give an optimal approximation to the state of each word, where "optimal" means that it gives the linear model of a given size, k that is best able to estimate labels that depend linearly on state, subject to only using the word and not its context. The right canonical correlates similarly give optimal state estimates given the context.

5. Due to the Zipfian nature of the word distribution, we will pretend that the means are all in fact zero and refer to these matrices as covariance matrices, when in fact they are second moment matrices.

OSCCA, as defined in Equation 2 thus gives an efficient way to calculate the eigenword dictionary $\boldsymbol{\phi}_w$ for a set of v words given the context and associated word matrices from a corpus.

3.1.1 Theoretical Properties

We now discuss how well the hidden state can be estimated from the target word. (A similar result can be derived for estimating hidden state from the context.) The state estimated is arbitrary up to any linear transformation, so all our comments address our ability to use the state to estimate some label which depends linearly on the state.

Keeping the dominant singular vectors in $\boldsymbol{\phi}_w$ and $\boldsymbol{\phi}_c$ provides two different bases for the estimated state. Each is optimal in its own way, as explained below.

The following Theorem 1 shows that the left canonical correlates give an optimal approximation to the state of each word (in the sense of being able to estimate an emission or label y for each state), subject to only using the word and not its context.

Theorem 1 *Let $\{w_i, c_i, y_i\}$ $(\in \mathbb{R}^v \times \mathbb{R}^{hv} \times \mathbb{R})$ for $i = 1 \dots n$ be n observations of random variables drawn i.i.d. from some distribution (pdf or pmf) $\mathbb{D}(w, c, y)$. We call the pair $(y_1 \dots y_n, \beta)$ a* linear context problem *if*

1. *y_i is a linear function of the context (i.e. $y_i = \alpha^\top c_i$).*
2. *$\beta^\top w_i$ is the best linear estimator of y_i given w_i, namely β minimizes $\sum_{i=1}^n (y_i - \beta^\top w_i)^2$ and*
3. *$Var(y_i) \leq 1$.*

Let $(\boldsymbol{\phi}_w, \boldsymbol{\phi}_c) \equiv CCA(\mathbf{W}, \mathbf{C})$ where $\mathbf{W}$ and $\mathbf{C}$ are the matrices constructed from $\{w\}_{i=1}^n$ and $\{c\}_{i=1}^n$ respectively. Also, let ${\boldsymbol{\phi}_w}^i$ be the i^{th} left singular vector. Then, for all $\epsilon > 0$ there exists a k such that for any linear context problem $(y_1 \dots y_n, \beta)$, there exists a $\gamma \in \mathbb{R}^k$ such that $\widehat{y}_i = \sum_{j=1}^k \gamma_j \boldsymbol{\phi}_w^{ji}$ is a good approximation to y_i in the sense that $\sum_{i=1}^n (\widehat{y}_i - \beta^\top w_i)^2 \leq \epsilon$.

Please see Appendix A for the proof.

To understand the above theorem, note that we would have liked to have a linear regression predicting some label y from the original data w. However, the original data is very high ('v') dimensional. Instead, we can first use CCA to map high dimensional vectors w to lower dimensional vectors $\boldsymbol{\phi}_w$, from which y can be predicted. For example with a few labeled examples of the form (w, y), we can recover the γ_i parameters using linear regression. The $\boldsymbol{\phi}_w$ subspace is guaranteed to hold a good approximation. A special case of interest occurs when estimating a label z $(= \alpha^\top c)$ plus zero mean noise. In this case, one can pick $y = \mathbb{E}(z)$ and proceed as above. This effectively extends the theorem to the case where the mapping from c to y is random, not deterministic.

Note that if we had used covariance rather than correlation as done by LSA/PCA then in the worst case, the key singular vectors for predicting state could be those with arbitrarily small singular values. This corresponds to the fact that for principle component regression (PCR), there is no guarantee that the largest principle components will prove predictive of an associated label.

One can think of Theorem 1 as implicitly estimating a k-dimensional hidden state from the observed w. This hidden state can be used to estimate y. Note that for Theorem 1, the state estimate is "trivial" in the sense that because it comes from the words, not the context, every occurrence of each word must give the same state estimate. This is attractive in that it associates a latent vector with every word type, but limiting in that it does not allow for any word ambiguity. The right canonical vectors allow one to estimate state from the context of a word, giving different state estimates for the same word in different contexts, as is needed for word sense disambiguation. We relegate that discussion to later in the paper, when we discuss induction of context-specific word embeddings. For now, we focus on the simpler use of left canonical covariates to map each word type to a k dimensional vector.

4. Efficient Eigenwords with Better Sample Complexity

OSCCA is optimal only in the limit of infinite data. In practice, data is, of course, always limited. In languages, lack of data comes about in two ways. Some languages are resource poor; one just does not have that many tokens of them (especially languages that lack a significant written literature). Even for most modern languages, many of the individual words in them are quite rare. Due to the Zipfian distribution of words, many words do not show up very often. A typical year's worth of Wall Street Journal text only has "lasagna" or "backpack" a handful of times and "ziti" at most once or twice. To overcome these issues we propose a two-step procedure which gives rise to two algorithms, Two Step CCA (TSCCA) and Low-Rank Multi-View Learning (LR-MVL) that have better sample complexity for rare words.

4.1 Two Step CCA (TSCCA) for Estimating Eigenword Dictionary

We now introduce our two step procedure TSCCA of computing an *eigenword dictionary* and show theoretically that it gives better estimates than the OSCCA method described in the last section.

In the two-step method, instead of taking the CCA between the combined context [**L R**] and the words **W**, we first take the CCA between the left and right contexts and use the result of that CCA to estimate the state **S** (an empirical estimate of the true hidden state $\hbar$) of all the tokens in the corpus from their contexts. Note that we get partially redundant state estimates from the left context and from the right context; these are concatenated to make combined state estimate. This will contain some redundant information, but will not lose any of the differences in information from the left and right sides. We then take the CCA between **S** and the words **W** to get our final *eigenword dictionary*. This is summarized in Algorithm 1. The first step, the CCA between **L** and **R**, must produce at least as many canonical components as the second step, which produces the final output.

The two step method requires fewer tokens of data to get the same accuracy in estimating the *eigenword dictionary* because its final step estimates fewer parameters $O(vk)$ than the OSCCA does $O(v^2)$.

Before stating the theorem, we first explain this intuitively. Predicting each word as a function of all other word combinations that can occur in the context is far sparser than predicting low dimensional state from context, and then predicting word from state. Thus, for relatively infrequent words, OSCCA should have significantly lower accuracy than the

Algorithm 1 Two step CCA

1: **Input:** $\mathbf{L}, \mathbf{W}, \mathbf{R}$
2: $(\boldsymbol{\phi}_l, \boldsymbol{\phi}_r) = CCA(\mathbf{L}, \mathbf{R})$
3: $\mathbf{S} = [\mathbf{L}\boldsymbol{\phi}_l \;\; \mathbf{R}\boldsymbol{\phi}_r]$
4: $(\boldsymbol{\phi}_s, \boldsymbol{\phi}_w) = CCA(\mathbf{S}, \mathbf{W})$
5: **Output:** $\boldsymbol{\phi}_w$, the eigenword dictionary

two step version. Phrased differently, mapping from context to state and then from state to word (TSCCA) gives a more parsimonious model than mapping directly from context to word (OSCCA).

The relative ability of OSCCA to estimate hidden state compared to that of TSCCA can be summarized as follows:

Theorem 2 *Given a matrix of words,* $\mathbf{W}$ *and their associated left and right contexts,* $\mathbf{L}$ *and* $\mathbf{R}$ *with vocabulary size* v*, context size* h*, and corpus of* n *tokens. Consider a linear estimator built on the state estimates estimated by either TSCCA or OSCCA, then the ratio of their squared prediction errors (i.e. relative statistical efficiency) is* $\frac{h+k}{hv}$*.*

Please see Appendix A for the proof.

Since the corpora we care about (i.e. text and language corpora) usually have $vh \gg h + k$, the TSCCA procedure will in expectation correctly estimate hidden state with a much smaller number of components k than the one step procedure. Or, equivalently, for an estimated hidden state of given size k, TSCCA will correctly estimate more of the hidden state components.

As mentioned earlier, words have a Zipfian distribution so most words are rare. For such rare words, if one does a CCA between them and their contexts, one will have very few observations, and hence will get a low quality estimate of their eigenword vector. If, on the other hand, one first estimates a state vector for the rare words, and then does a CCA between this state vector and the context, the rare words can be thought of as borrowing strength from more common distributionally similar words. For example, "umbrage" (56,020) vs. "annoyance" (777,061) or "unmeritorious" (9,947) vs. "undeserving" (85,325). The numbers in parentheses are the number of occurrences of these words in the Google n-gram collection used in some of our experiments.

4.2 Low Rank Multi-View Learning (LR-MVL)

The context around a word, consisting of the h words to the right and left of it, sits in a high dimensional space, since for a vocabulary of size v, each of the h words in the context requires an indicator function of dimension v. So, we propose an algorithm Low Rank Multi-View Learning (LR-MVL), where we work in the k dimensional space to begin with.

The key move in LR-MVL is to project the hv-dimensional $\mathbf{L}$ and $\mathbf{R}$ matrices down to a k dimensional state space before performing the first CCA. This is where it differs from TSCCA. Thus, all eigenvector computations are done in a space that is v/k times smaller than the original space. Since a typical vocabulary contains at least $100,000$ words, and we

use state spaces of order $k \approx 100$ dimensions, this gives a 1,000-fold reduction in the size of calculations that are needed.

LR-MVL iteratively updates the real-valued state of a token $\mathbf{Z_t}$, till convergence. Since, the state is always real-valued, this also allows us to replace the projected left and right contexts with exponential smooths (weighted average of the previous (or next) token's state i.e. $\mathbf{Z_{t-1}}$ (or $\mathbf{Z_{t+1}}$) and previous (or next) token's smoothed state i.e. $\mathbf{S_{t-1}}$ (or $\mathbf{S_{t+1}}$).), of them at a few different time scales. One could use a mixture of both very short and very long contexts which capture short and long range dependencies as required by NLP problems as NER, Chunking, WSD etc. Since exponential smooths are linear, we preserve the linearity of our method.

We now describe the LR-MVL algorithms.

4.2.1 The LR-MVL Algorithms

Based on our theory (described in next subsection), various algorithms are possible for LR-MVL. We provide two algorithms, Algorithms 2, 3 (without and with exponential smooths).

Algorithm 2 LR-MVL Algorithm - Learning from Large amounts of Unlabeled Data (no exponential smooths).

1: **Input:** Token sequence $\mathbf{W}_{n\times v}$, state space size k.
2: Initialize the eigenfeature dictionary $\boldsymbol{\phi}_w$ to random values $\mathcal{N}(0,1)$.
3: **repeat**
4: Project the left and right context matrices $\mathbf{L}_{n\times vh}$ and $\mathbf{R}_{n\times vh}$ down to 'k' dimensions and compute CCA between them. $[\boldsymbol{\phi}_l, \boldsymbol{\phi}_r]$=CCA($\mathbf{L}\boldsymbol{\phi}^{\mathbf{h}}_{\mathbf{w}}$, $\mathbf{R}\boldsymbol{\phi}^{\mathbf{h}}_{\mathbf{w}}$). //$\boldsymbol{\phi}^h_w$ is the stacked version of $\boldsymbol{\phi}_w$ matrix as many times as the context length 'h.'
5: Normalize $\boldsymbol{\phi}^{(k)}_l$ and $\boldsymbol{\phi}^{(k)}_r$. //Divide each row by the maximum absolute value in that row (Scales between -1 and +1).
6: Compute a second CCA between the estimated state and the word itself $[\boldsymbol{\phi}_w, \boldsymbol{\phi}_c]$=CCA($\mathbf{W}$, $[\mathbf{L}\boldsymbol{\phi}^{\mathbf{h}}_{\mathbf{w}}\boldsymbol{\phi}^{(k)}_l$, $\mathbf{R}\boldsymbol{\phi}^{\mathbf{h}}_{\mathbf{w}}\boldsymbol{\phi}^{(k)}_r]$).
7: Compute the change in $\boldsymbol{\phi}_w$ from the previous iteration
8: **until** $|\Delta\boldsymbol{\phi}^h_w| < \epsilon$
9: **Output:** $\boldsymbol{\phi}_l$, $\boldsymbol{\phi}_r$, $\boldsymbol{\phi}_w$.

A few iterations (~ 10) of the above algorithms are sufficient to converge to the solution[6].

4.2.2 Theoretical Properties of LR-MVL

We now present the theory behind the LR-MVL algorithms; particularly we show that the reduced rank matrix $\boldsymbol{\phi}_w$ allows a significant data reduction while preserving the information in our data and the estimated state does the best possible job of capturing any label information that can be inferred by a linear model.

The key difference from TSCCA is that we can initialize the state of each word randomly and work in a low (k) dimensional space from the beginning, iteratively refine the state till convergence and still we can recover the eigenword dictionary $\boldsymbol{\phi}_w$.

6. Though the optimization problem and our iterative procedure are non-convex, empirically we did not face any issues with convergence.

Algorithm 3 LR-MVL Algorithm - Learning from Large amounts of Unlabeled Data (with exponential smooths).

1: **Input:** Token sequence $\mathbf{W}_{n\times v}$, state space size k, smoothing rates α^j
2: Initialize the eigenfeature dictionary $\boldsymbol{\phi}_w$ to random values $\mathcal{N}(0,1)$.
3: **repeat**
4: Set the state Z_t $(1 < t \leq n)$ of each token w_t to the eigenword vector of the corresponding word.
$Z_t = (\boldsymbol{\phi}_w : w = w_t)$
5: Smooth the state estimates before and after each token to get a pair of views for each smoothing rate α^j.
$S_t^{(l,j)} = (1-\alpha^j)S_{t-1}^{(l,j)} + \alpha^j Z_{t-1}$ // left view $\mathbf{L}$
$S_t^{(r,j)} = (1-\alpha^j)S_{t+1}^{(r,j)} + \alpha^j Z_{t+1}$ // right view $\mathbf{R}$.
where the t^{th} rows of $\mathbf{L}$ and $\mathbf{R}$ are, respectively, concatenations of the smooths $S_t^{(l,j)}$ and $S_t^{(r,j)}$ for each of the $\alpha^{(j)}$s.
6: Find the left and right canonical correlates, which are the eigenvectors $\boldsymbol{\phi}_l$ and $\boldsymbol{\phi}_r$ of
$(\mathbf{L}^\top\mathbf{L})^{-1}\mathbf{L}^\top\mathbf{R}(\mathbf{R}^\top\mathbf{R})^{-1}\mathbf{R}^\top\mathbf{L}\boldsymbol{\phi}_l = \lambda\boldsymbol{\phi}_l$.
$(\mathbf{R}^\top\mathbf{R})^{-1}\mathbf{R}^\top\mathbf{L}(\mathbf{L}^\top\mathbf{L})^{-1}\mathbf{L}^\top\mathbf{R}\boldsymbol{\phi}_r = \lambda\boldsymbol{\phi}_r$.
7: Project the left and right views on to the space spanned by the top k left and right CCAs respectively
$\mathbf{X_l} = \boldsymbol{L}\boldsymbol{\phi}_l^{(k/2)}$ and $\mathbf{X_r} = \boldsymbol{R}\boldsymbol{\phi}_r^{(k/2)}$
where $\boldsymbol{\phi}_l^{(k)}$, $\boldsymbol{\phi}_r^{(k)}$ are matrices composed of the singular vectors of $\boldsymbol{\phi}_l$, $\boldsymbol{\phi}_r$ with the k largest magnitude singular values. Estimate the state for each word w_t as the union of the left and right estimates: $\mathbf{Z} = [\mathbf{X_l},\ \mathbf{X_r}]$
8: Compute a second CCA between the estimated state and the word itself $[\boldsymbol{\phi}_w, \boldsymbol{\phi}_z]$=CCA($\mathbf{W}$, $\mathbf{Z}$).
9: Normalize $\boldsymbol{\phi}_w$. //Divide each row by the maximum absolute value in that row (Scales between -1 and +1).
10: Compute the change in $\boldsymbol{\phi}_w$ from the previous iteration.
11: **until** $|\Delta\boldsymbol{\phi}_w| < \epsilon$
12: **Output:** $\boldsymbol{\phi}_l^k$, $\boldsymbol{\phi}_r^k$, $\boldsymbol{\phi}_w$.

As earlier, let $\mathbf{L}$ be an $n \times hv$ matrix giving the words in the left context of each of the n tokens, where the context is of length h, $\mathbf{R}$ be the corresponding $n \times hv$ matrix for the right context, and $\mathbf{W}$ be an $n \times v$ matrix of indicator functions for the words themselves. Note that $\mathbf{L}$, $\mathbf{R}$ and $\mathbf{W}$ are the observed instantiations of the corresponding multivariate random variables l, r and w.

The theory of LR-MVL hinges on four assumptions which are described in detail in the appendix. Basically, they entail that there exists a k dimensional linear hidden state for l, r and w and that they come from a HMM with rank k observation and transition matrices. It's further assumed that the pairwise expected correlations between l, r and w, also have rank k.

Lemma 3 *Define $\boldsymbol{\phi}_w$ as the left singular vectors:*

$$\boldsymbol{\phi}_w \equiv CCA(w, [l \quad r])_{left}.$$

where CCA(z, w) is defined as in Equation 1 but using population covariance matrices i.e. $\mathbf{C}_{zw} = \mathbb{E}(z^\top w)$, $\mathbf{C}_{zz} = \mathbb{E}(z^\top z)$ *and* $\mathbf{C}_{ww} = \mathbb{E}(w^\top w)$.
Under assumptions 2, 3 and 1A(in appendix) such that if $(\boldsymbol{\phi}_l, \boldsymbol{\phi}_r) \equiv CCA(l, r)$ *then*

$$\boldsymbol{\phi}_w = CCA(w, [l\phi_l \quad r\phi_r])_{left}.$$

Please see Appendix A for the proof.

Lemma 3 shows that instead of finding the CCA between the full context and the words, we can take the CCA between the Left and Right contexts, estimate a k dimensional state from them, and take the CCA of that state with the words and get the same result. Lemma 3 is similar to Theorem 2, except that it does not provide ratios of the estimated state sizes.

Let $\boldsymbol{\phi}_w^h$ denote a matrix formed by stacking h copies of $\boldsymbol{\phi}_w$ on top of each other. Right multiplying l or r by $\boldsymbol{\phi}_w^h$ projects each of the words in that context into the k-dimensional reduced rank space.

The following theorem addresses the core of the LR-MVL algorithm, showing that there is an $\boldsymbol{\phi}_w$ which gives the desired dimensionality reduction. Specifically, it shows that the previous lemma also holds in the reduced rank space.

Theorem 4 *Under assumptions 1, 1A and 2 (in appendix) there exists a unique matrix $\boldsymbol{\phi}_w$ such that if*

$$(\boldsymbol{\phi}_l^h, \boldsymbol{\phi}_r^h) \equiv CCA(l\boldsymbol{\phi}_w^h, r\boldsymbol{\phi}_w^h),$$

then

$$\boldsymbol{\phi}_w = CCA(w, [l\boldsymbol{\phi}_w^h\boldsymbol{\phi}_l^h \quad r\boldsymbol{\phi}_w^h\boldsymbol{\phi}_r^h])_{left},$$

where $\boldsymbol{\phi}_w^h$ is the stacked form of $\boldsymbol{\phi}_w$.

Please see Appendix A for the proof[7].

Because of the Zipfian distribution of words, many words are rare or even unique. So, just as in the case of TSCCA, CCA between the rare words and context will not be informative, whereas finding the CCA between the projections of left and right contexts gives a good state vector estimate even for unique words. One can then fruitfully find the CCA between the contexts and the estimated state vector for their associated words.

5. Generating Context Specific Embeddings

Once we have estimated the CCA model using any of our algorithms (i.e. OSCCA, TSCCA, LR-MVL), it can be used to generate context specific embeddings for the tokens from training, development and test sets (as described in Algorithm 4). These embeddings could be

7. It is worth noting that our matrix $\boldsymbol{\phi}_w$ corresponds to the matrix $\hat{\mathbf{U}}$ used by (Hsu et al., 2009; Siddiqi et al., 2010). They showed that $\mathbf{U}$ is sufficient to compute the probability of a sequence of words generated by an HMM, our $\phi_{\mathbf{w}}$ provides a more statistically efficient estimate of $\mathbf{U}$ than their $\hat{\mathbf{U}}$, and hence can also be used to estimate the sequence probabilities.

further supplemented with other baseline features and used in a supervised learner to predict the label of the token.

Algorithm 4 Inducing Context Specific Embeddings for Train/Dev/Test Data

1: **Input:** Model ($\boldsymbol{\phi}_l^k$, $\boldsymbol{\phi}_r^k$, $\boldsymbol{\phi}_w$) output from above algorithm and Token sequences $\mathbf{W^{train}}$, ($\mathbf{W^{dev}}$, $\mathbf{W^{test}}$)
2: Project the left and right views $\mathbf{L}$ and $\mathbf{R}$ onto the space spanned by the top k left and right CCAs respectively. If algorithm is Algorithm 3, then, smooth $\mathbf{L}$ and $\mathbf{R}$ first.
$\mathbf{X_l} = \boldsymbol{L}\boldsymbol{\phi}_l^k$ and $\mathbf{X_r} = \boldsymbol{R}\boldsymbol{\phi}_r^k$
and the words onto the eigenfeature dictionary $\mathbf{X_w} = \boldsymbol{W}^{train}\boldsymbol{\phi}_w$
3: Form the final embedding matrix $\mathbf{X_{train:embed}}$ by concatenating these three estimates of state
$\mathbf{X_{train:embed}} = [\mathbf{X_l}\ \ \mathbf{X_w}\ \ \mathbf{X_r}]$
4: **Output:** The embedding matrices $\mathbf{X_{train:embed}}$, ($\mathbf{X_{dev:embed}}$, $\mathbf{X_{test:embed}}$) with context-specific representations for the tokens.

Note that we can get context "oblivious" embeddings i.e. one embedding per word type, just by using the eigenfeature dictionary $\boldsymbol{\phi}_w$. Later in the experiments section we show that this approach of inducing context specific embeddings gives results which are similar to a simpler alternative of just using the context "oblivious" embeddings but augmenting them with the embeddings of the words in a window of 2 around the current word before using them in a classifier.

6. Efficient Estimation

As mentioned earlier, CCA can be done by taking the singular value decomposition of a matrix. For small matrices, this can be done using standard functions in e.g. MATLAB, but for very large matrices (e.g. for vocabularies of tens or hundreds of thousands of words), it is important to take advantage of the recent advances in SVD algorithms. For our experiments we use the method of (Halko et al., 2011), which uses random projections to compute SVD of large matrices.

The key idea is to find a lower dimensional basis for $\mathbf{A}$, and to then compute the singular vectors in that lower dimensional basis. The initial basis is generated randomly, and taken to be slightly larger than the eventual basis. If $\mathbf{A}$ is $v \times hv$, and we seek a state of dimension k, we start with a $hv \times (k+l)$ matrix $\mathbf{\Omega}$ of random numbers, where l is number of "extra" basis vectors between 0 and k. We then project $\mathbf{A}$ onto this matrix and take the SVD decomposition of the resulting matrix ($\mathbf{A} \approx \hat{\mathbf{U}}\hat{\mathbf{\Lambda}}\hat{\mathbf{V}}^\top$).

Since $\mathbf{A\Omega}$ is $v \times (k+l)$, this is much cheaper than working on the original matrix $\mathbf{A}$. We keep the largest k components of $\mathbf{U}$ and of $\mathbf{V}$, which form a left and a right basis for $\mathbf{A}$ respectively.

This procedure is repeated for a few (~ 5) iterations. The algorithm is summarized in Algorithm 5. The runtime of the procedure for projecting a matrix of size $m \times p$ down to a size $m \times k$ where $p \gg k$ is O(mpk) floating point operations, which in our case becomes O(v^2hk).

(Halko et al., 2011) prove a number of nice properties of the above algorithm. In particular, they guarantee that the algorithm, even without the extra iterations in steps 3 and 6 produces an approximation whose error is bounded by a small polynomial factor times the

Algorithm 5 Randomized singular value decomposition

Input: Matrix $\mathbf{A}$ of size $v \times hv$, the desired hidden state dimension k, and the number of "extra" singular vectors, l
Generate a $hv \times (k+l)$ random matrix $\mathbf{\Omega}$
for i =1:5 **do**
  $\mathbf{M} = \mathbf{A\Omega}$.
  $[\mathbf{Q}, \mathbf{R}]$=QR($\mathbf{M}$) //Find $v \times (k+l)$ orthogonal matrix $\mathbf{Q}$.
  $\mathbf{B} = \mathbf{Q}^\top \mathbf{A}$
  $\mathbf{\Omega} = \mathbf{B}$
end for
Find the SVD of $\mathbf{B}$. $[\hat{\mathbf{U}}, \hat{\mathbf{\Lambda}}, \hat{\mathbf{V}}^\top]$ =SVD($\mathbf{B}$), and keep the k components of $\hat{\mathbf{U}}$ with the largest singular values.
$\tilde{\mathbf{A}} = \mathbf{Q}\hat{\mathbf{U}}$. //Compute the rank-k projection.
Output: The rank-k approximation $\tilde{\mathbf{A}}$. (Similar procedure can be repeated to get the right singular values and the corresponding projections.)

size of the largest singular value whose singular vectors are *not* part of the approximation, σ_{k+1}. They also show that using a small number of "extra" singular vectors (l) results in a substantial tightening of the bound, and that the extra iterations, which correspond to power iteration, drive the error bound exponentially quickly to one times the largest non-included singular value, σ_{k+1} and also provide better separation between the singular values.

7. Evaluating Eigenwords

In this section we provide qualitative and quantitative evaluation of the various eigenword algorithms.

The state estimates for words capture a wide range of information about them that can be used to predict part of speech, linguistic features, and meaning. Before presenting a more quantitative evaluation of predictive accuracy, we present some qualitative results showing how word states, when projected in appropriate directions usefully characterize the words.

We compare our approach against several state-of-the-art word embeddings:

1. Turian Embeddings (C&W and HLBL) (Turian et al., 2010).

2. SENNA Embeddings (Collobert et al., 2011).

3. word2vec Embeddings (Mikolov et al., 2013a,b).

We also compare against simple PCA/LSA embeddings and other model based approaches wherever applicable.

We downloaded the Turian embeddings (C&W and HLBL), from `http://metaoptimize.com/projects/wordreprs` and use the best 'k' reported in the paper (Turian et al., 2010) i.e. k=200 and 100 respectively. SENNA embeddings were downloaded from `http://ronan.collobert.com/senna/`. word2vec code was downloaded from `https://code.google.com/p/word2vec/`. Since they made the code available we could train them on the exact same

corpora, had the exact same context window and vocabulary size as the eigenword embeddings. The PCA baseline used is similar to the one that has recently been proposed by (Lamar et al., 2010) except that here we are interested in supervised accuracy and not the unsupervised accuracy as in that paper.

In the results presented below (qualitative and quantitative), we trained all the algorithms (including eigenwords) on Reuters RCV1 corpus (Rose et al., 2002) for uniformity of comparison[8]. Case was left intact and we did not do any other "cleaning" of data. Tokenization was performed using NLTK tokenizer (Bird and Loper, 2004). RCV1 corpus contains Reuters newswire from Aug '96 to Aug '97 and containing about 215 million tokens after tokenization.

Unless otherwise stated, we consider a fixed window of two words (h=2) on either side of a given word and a vocabulary of 100,000 most frequent words for all the algorithms[9], in order to ensure fairness of comparison.

Eigenword algorithms are robust to the dimensionality of hidden space (k), so we did not tune it and fixed it at 200. For other algorithms, we report results using their best hidden space dimensionality.

Our theory and CCA in general (Bach and Jordan, 2005) rely on normality assumptions[10], however the words follow Zipfian (heavy tailed) distribution. So, we took the square root of the word counts in the context matrices (i.e. $\mathbf{W}^\top\mathbf{C}$) before running OSCCA, TSCCA and LR-MVL(I). This squishes the word distributions and makes them look more normal (Gaussian). This idea is not novel and dates back in statistics to Anscombe Transform (Anscombe, 1948) and has precedents even in word representation learning literature (Turney and Pantel, 2010).

We ran LR-MVL(I) and LR-MVL(II) for 5 iterations and only used one exponential smooth of 0.5 for LR-MVL(II). Table 1 shows the details of all the embeddings used in our experiments.

8. Qualitative Evaluation of OSCCA

To illustrate the sorts of information captured in our state vectors, we present a set of figures constructed by projecting selected small sets of words onto the space spanned by the second and third largest principal components of their eigenword dictionary values, which are simply the left canonical correlates calculated from Equation 2. (The first principle component generally just separates the selected words from other words, and so is less interesting here.)

Figure 3 shows plots for three different sets of words. The left column uses the eigenword dictionary learned using OSCCA (CCA($\mathbf{W}$, $\mathbf{C}$), where $\mathbf{C}$=[$\mathbf{L}$ $\mathbf{R}$] with h=2 on either side) (the other eigenword algorithms gave similar results), while the right column uses the corresponding latent vectors derived using PCA on the same data. In all cases, the 200-

8. word2vec, PCA and Turian (C&W and HLBL) embeddings are all trained on Reuters RCV1, but SENNA embeddings (training code not available) were trained on a larger Wikipedia corpus.
9. Turian (C&W and HLBL), SENNA embeddings had much bigger vocabulary sizes of 268,000 and 130,000, though they also use a window of 2 as context.
10. CCA can be thought of as least squares regression (Please see the proof of Theorem 2 in Appendix A.) and hence has error terms distributed normally.

Embedding	Unlabeled Data Trained	Window size	Vocab. Size	Hidden State Size	Availability
C&W (Turian)	Reuters RCV1 cleaned and lowercased (See Turian et al. 2010.)	2	268,810	200	Only Embeddings available (No Code).
HLBL (Turian)	Reuters RCV1 cleaned and lowercased (See Turian et al. 2010.)	2	268,810	100	Only Embeddings available (No Code).
SENNA	Wikipedia (much larger than RCV1) (See Collobert et al. 2011.)	2	130,000	50	Only Embeddings available (No Code).
Word2vec (SK-Continuous Skip-gram) & (CB-Continuous Bag-of-words)	Reuters RCV1 uncleaned and case intact	2	100,000	200	Code available
Eigenwords	Reuters RCV1 uncleaned and case intact	2	100,000	200	Code and Embeddings available

Table 1: Details of various embeddings used in the experiments. Note: Eigenwords and Word2vec provide the most controlled comparison.

dimensional vectors have been projected onto two dimensions (using a second PCA) so that they can be visualized.

The PCA algorithm differs from CCA based (eigenword) algorithms in that it does not whiten the matrices via ($\mathbf{C}_{zz}^{-1/2}$ and $\mathbf{C}_{ww}^{-1/2}$) before performing SVD. In other words, the PCA algorithm just operates on $\mathbf{W}^\top\mathbf{C}$. If one considers a word and its two grams to the left and right as a document, then its equivalent to the Latent Semantic Analysis (LSA) algorithm.

The results for various (handpicked) semantic categories are shown in Figure 3 and 4.

The top row shows a small set of randomly selected nouns and verbs. Note that for CCA, nouns are on the left, while verbs are on the right. Words that are of similar or opposite meaning (e.g. "agree" and "disagree") are distributionally similar, and hence close. The corresponding plot for PCA shows some structure, but does not give such a clean separation. This is not surprising; predicting the part of speech of words depends on the exact order of the words in their context (as we capture in CCA); a PCA-style bag-of-words can't capture part of speech well.

The bottom row in Figure 3 shows names of numbers or the numerals representing numbers and years. Numbers that are close to each other in value tend to be close in the

Center Word	OSCCA NN	PCA NN
market	markets, trade, currency, sector, activity.	dollar, economy, government, sector, industry.
company	firm, group, giant, operator, maker.	government, group, dollar, following, firm.
Ltd	Limited, Bhd, Plc, Co, Inc.	Corp, Plc, Inc, name, system.
President	Governor, secretary, Chairman, leader, Director.	Commerce, General, fuel, corn, crude.
Nomura	Daiwa, UBS, HSBC, NatWest, BZW.	Chrysler, Sun, Delta, Bre-X, Renault.
jump	drop, fall, rise, decline, climb.	surge, stakes, slowdown, participation, investing.
rupee	peso, zloty, crown, pound, franc.	crown, CAC-40, FTSE, Nikkei, 30-year.

Table 2: Nearest Neighbors of OSCCA and PCA word embeddings.

plot, thus suggesting that state captures not just classifications, but also more continuous hidden features.

The plots in Figure 4 show a similar trend i.e., eigenword embeddings are able to provide a clear separation between different syntactic/semantic categories and capture a rich set of features characterizing the words, whereas PCA mostly just squishes them together.

Table 2 shows the five nearest neighbors for a few representative words using OSCCA and PCA. As can be seen, the OSCCA based nearest neighbors capture subtle semantic and syntactic cues e.g Japanese investment bank (Nomura) having another Japanese investment bank (Daiwa) as the nearest neighbor, whereas the PCA nearest neighbors are more noisy and capture mostly syntactic aspects of the word.

9. Quantitative Evaluation

This section describes the performance (accuracy and richness of representation) of various eigenword algorithms. We evaluate the quality of the *eigenword dictionary* by using it in a supervised learning setting to predict a wide variety of labels that can be attached to words.

We perform experiments for a variety of NLP tasks including, Word Similarity, Sentiment Classification, Named Entity Recognition (NER), chunking, Google semantic and syntactic analogy tasks and Word Sense Disambiguation (WSD) to demonstrate the richness of the state learned by eigenwords and that they perform comparably or better than other state-of-the-art approaches. For these tasks, we report results using the best eigenwords for compactness, though all the four algorithms gave similar performances.

However, before we proceed to do that, we compare OSCCA against TSCCA, LR-MVL(I) and LR-MVL(II) embeddings on a set of Part of Speech (POS) tagging problems for different languages, looking at how the predictive accuracy scales with corpus size for predictions on a fixed vocabulary. These results use small corpora and demonstrate that TSCCA, LR-MVL(I) and LR-MVL(II) perform better for rarer words.

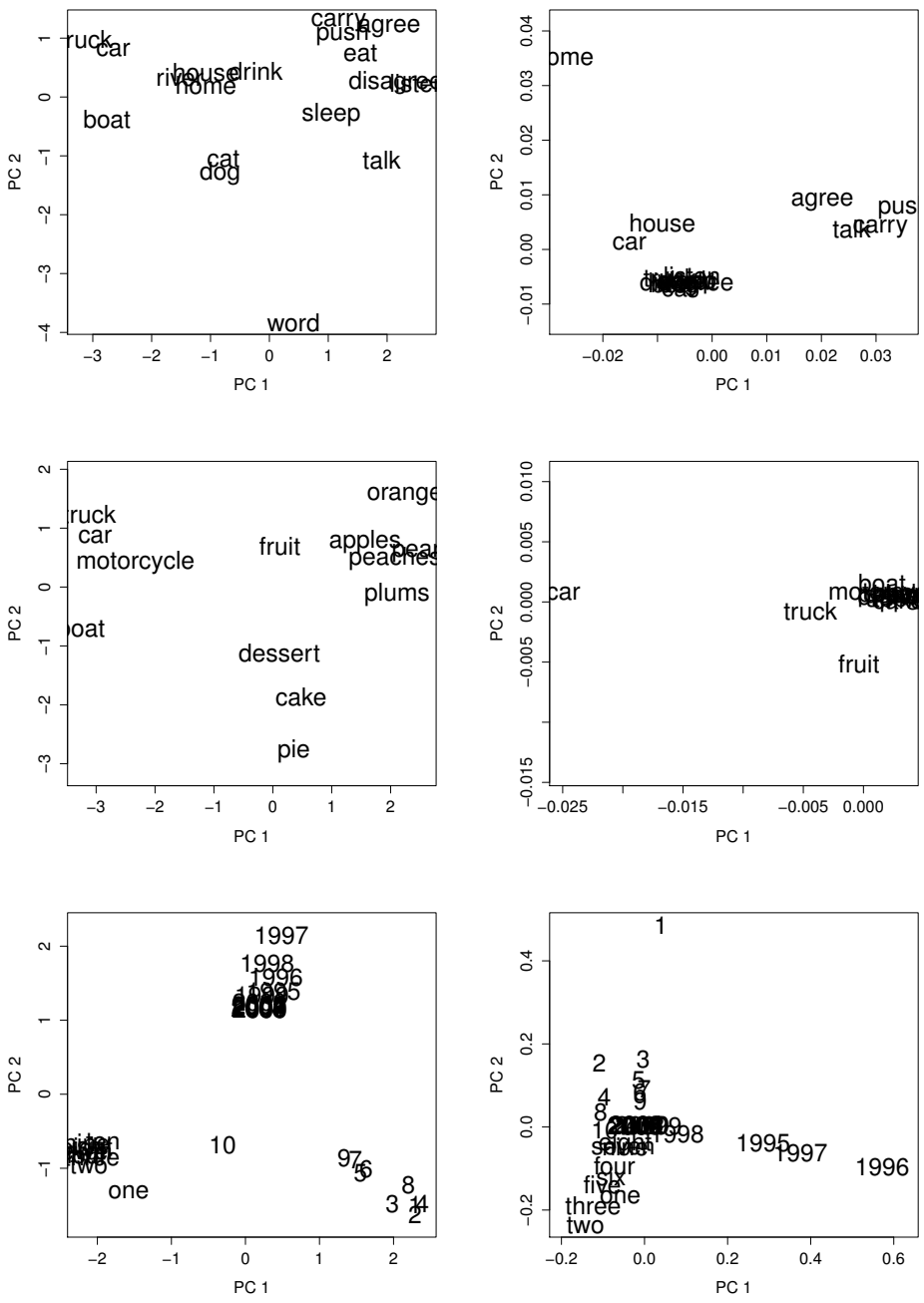


Figure 3: Projections onto two dimension of selected words in different categories using both OSCCA (left) and PCA (Right). Top to bottom: 1). (Nouns vs Verbs): house, home, dog, truck, boat, word, river, cat, car, sleep, eat, push, drink, listen, carry, talk, disagree, agree. 2). (Eateries vs vehicles): apples, pears, plums, oranges, peaches, fruit, cake, pie, dessert, truck, boat, car, motorcycle. 3). (Numerals vs letter numbers vs years): one, two, three, four, five, six, seven, eight, nine, ten, 1, 2,. . ., 10, 1995, 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009.

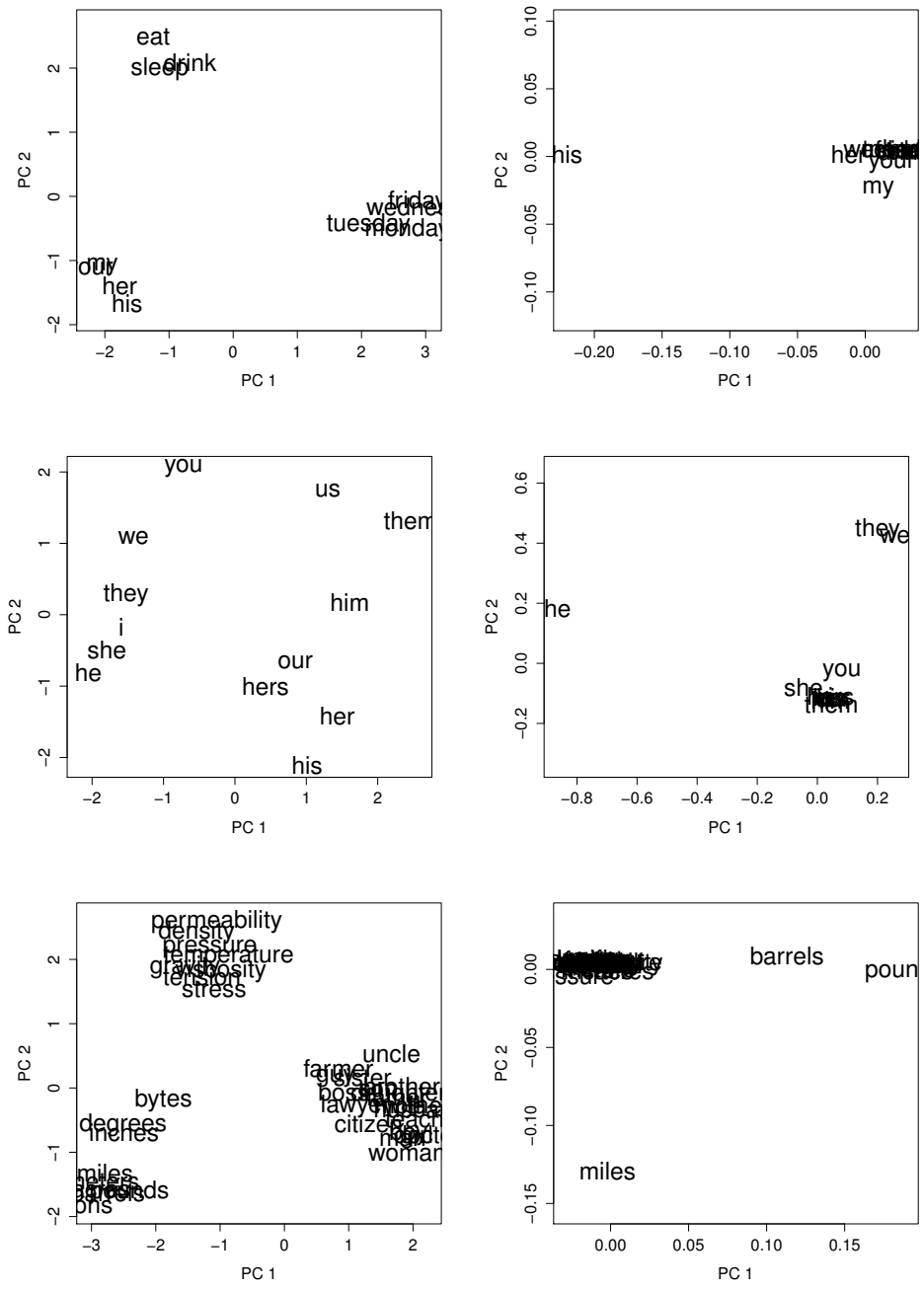


Figure 4: Projections onto two dimension of selected words in different categories using both OSCCA (left) and PCA (Right). Top to bottom: 1). (Weekdays vs verbs vs pronouns): monday, tuesday, wednesday, sunday, friday, eat, drink, sleep, his, her, my, your. 2). (Different kinds of pronouns): i, you, he, she, they, we, us, them, him, her, our, his, hers. 3). (Nouns vs Adjectives vs Units of measurement): man, woman boy, girl, lawyer, doctor, guy, farmer, teacher, citizen, mother, wife, father, son, husband, brother, daughter, sister, boss, uncle, pressure, temperature, permeability, density, stress, viscosity, gravity, tension, miles, pounds, degrees, inches, barrels, tons, acres, meters, bytes.

Language	Number of POS tags	Number of tokens
English	17	100311
Danish	25	100238
Bulgarian	12	100489
Portuguese	22	100367

Table 3: Description of the POS tagging data sets

9.1 Part of Speech (POS) Tagging

In this experiment we compare the performance of various eigenword algorithms on the task of non-disambiguating POS tagging for four languages; i.e., each word type has a single POS tag. Table 2 provides statistics on all the corpora used, namely: the Wall Street Journal portion of the Penn treebank (Marcus et al., 1993) (we consider the 17 tags of (PTB 17) (Smith and Eisner, 2005)), the Bosque subset of the Portuguese Floresta Sinta(c)tica Treebank (Afonso et al., 2002), the Bulgarian BulTreeBank (Simov et al., 2002) (with only the 12 coarse tags), and the Danish Dependency Treebank (DDT) (Kromann, 2003).

Note the corpora range widely in size; English has ~ 1 million tokens whereas Danish only has $\sim 100k$ tokens. To address this data imbalance we kept only the first $\sim 100k$ tokens of the larger corpora so as to perform a uniform evaluation across all corpora.

The goal of this experiment is see to how the eigenword dictionary estimates for the word types (for a fixed vocabulary) improve with increased training data.

Theorem 2 implies that the difference between OSCCA and TSCCA/LR-MVL(I)/LR-MVL(II) should be more pronounced at smaller sample sizes, where the errors are higher and that they should have similar predictive power in the limit of large training data. We therefore evaluate the performance of the methods on varying data sizes ranging from $5k$ to the entire $100k$ tokens.

When varying the unlabeled data from $5k$ to $100k$ we made sure that they had the exact same vocabulary to assure that the performance improvement is not coming from word types not present in the $5k$ tokens but present in the total $100k$. This gives a clear picture of the effect of varying training set size.

To evaluate the predictive accuracy of the descriptors learned using different amounts of unlabeled data, we learn a multi-class logistic regression to predict the POS tag of each type. We trained using 80% of the word types chosen randomly and then tested on the remaining 20% types. This procedure was repeated 10 times. Note that our train and test sets do not contain any of the same word types[11].

The accuracy of using OSCCA, TSCCA, LR-MVL(I), LR-MVL(II) and PCA features in a supervised learner are shown in Figure 5 for the task of POS tagging. As can be seen from the results, eigenword embeddings are significantly better (5% significance level in a paired t-test) than the PCA-based supervised learner. Among the eigenwords, TSCCA, LR-MVL(I) and LR-MVL(II) are significantly better than OSCCA for small amounts of data and, as predicted by theory, the two become comparable in accuracy as the amount of unlabeled data used to learn the CCAs becomes large.

11. As noted, we are doing non-disambiguating POS tagging so that each word type has a single POS tag, so if the same word type occurred in both the training and testing data, a learning algorithm that just memorized the training set would perform reasonably well.

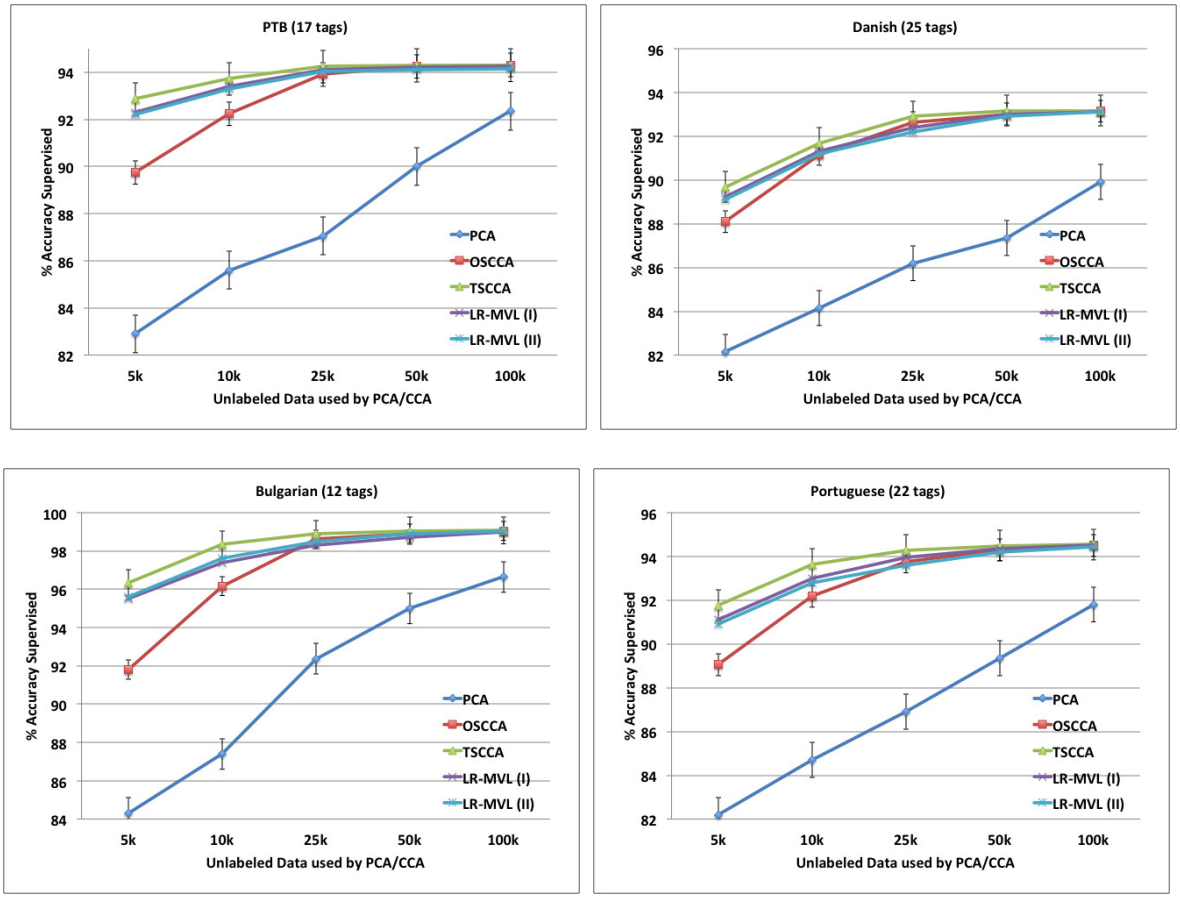


Figure 5: Plots showing accuracy as a function of number of tokens used to train the PCA/eigenwords for various languages. **Note:** The results are averaged over 10 random, 80 : 20 splits of word types.

9.2 Word Similarity Task (WordSim-353)

A standard data set for evaluating vector-space models is the WordSim-353 data set (Finkelstein et al., 2001), which consists of 353 pairs of nouns. Each pair is presented without context and associated with 13 to 16 human judgments on similarity and relatedness on a scale from 0 to 10. For example, (professor, student) received an average score of 6.81, while (professor, cucumber) received an average score of 0.31.

For this task, it is interesting to see how well the cosine similarity between the word embeddings correlates with the human judgment of similarity between the same two words. The results in Table 4 show the Spearman's correlation between the cosine similarity of the respective word embeddings and the human judgments.

As can be seen, eigenwords are statistically significantly (computed using resampled bootstrap) better than all embeddings except SENNA.

Model	$\rho\times$ 100
PCA	30.25
Turian (C&W)	28.08
Turian (HLBL)	35.24
SENNA	44.32
word2vec (SK)	42.73
word2vec (CB)	42.97
eigenwords (OSCCA)	43.00
eigenwords (TSCCA)	**44.85**
eigenwords (LR-MVL(I))	43.83
eigenwords (LR-MVL(II))	37.92

Table 4: Table showing the Spearman correlation between the word embeddings based similarity and human judgment based similarity. Note that the numbers for word2vec are different from the ones reported elsewhere, which is due to the fact that we considered a 100,000 vocabulary and a context window of 2 just like eigenwords, in order to make a fair comparison.

9.3 Sentiment Classification

It is often useful to group words into semantic classes such as colors or numbers, professionals or disciplines, happy or sad words, words of encouragement or discouragement, and, of course, words indicating positive or negative sentiment. Substantial effort has gone into creating hand-curated words that can be used to capture a variety of opinions about different products, papers, or people. To pick one example, (Teufel, 2010) contains dozens of carefully constructed lists of words that she uses to categorize what authors say about other scientific papers. Her categories include "problem nouns" (caveat, challenge, complication, contradiction,...), "comparison nouns" (accuracy, baseline, comparison, evaluation,...), "work nouns" (account, analysis, approach,...) as well as more standard sets of positive, negative, and comparative adjectives.

Psychologists, in particular, have created many such hand curated lists of words, such as the widely used LIWC collection (Pennebaker et al., 2001), which has a heterogeneous set of word lists ranging from "positive emotion" to "pronouns," "swear words" and "body parts." In the example below, we use words from a more homogeneous psychology collection, a set of five dimensions that have been identified in positive psychology under the acronym PERMA (Seligman, 2011):

- *Positive emotion* (aglow, awesome, bliss, ...),
- *Engagement* (absorbed, attentive, busy, ...),
- *Relationships* (admiring, agreeable, ...),
- *Meaning* (aspire, belong, ...)
- *Achievement* (accomplish, achieve, attain, ...).

Word sets	Number of observations	
	Class I	Class II
Positive emotion or not	81	162
Meaningful life or not	246	46
Achievement or not	159	70
Engagement or not	208	93
Relationship or not	236	204

Table 5: Description of the data sets used. All the data was collected from the PERMA lexicon.

For each of these five categories, we have both positive words – ones that connote, for example, *achievement*, and negative words, for example, *un-achievement* (amateurish, blundering, bungling, ...). We would hope (and we show below that this is in fact true), that we can use eigenwords not only to distinguish between different PERMA categories, but also to address the harder task of distinguishing between positive and negative terms in the same category. (The latter task is harder because words that are opposites, such as "large" and "small," often are distributionally similar.)

The description of the PERMA data sets is given in Table 5 and Table 6 shows results for the five PERMA categories. As earlier, we used logistic regression for the supervised binary classification.

As can be seen from the plots, the eigenwords perform significantly (5% significance level in a paired t-test) better than all other embeddings in 3/5 cases and for the remaining 2 cases they perform significantly better than all embeddings except word2vec.

9.4 Named Entity Recognition (NER) & Chunking

In this section we present the experimental results of eigenwords on Named Entity Recognition (NER) and chunking. For the previous evaluation tasks we were performing classification of individual words in isolation, however NER and chunking tasks involve assigning tasks to running text. This allows us to induce context specific embeddings i.e. a different embedding for a word based on its context.

9.4.1 Datasets and Experimental Setup

For the NER experiments we used the data from CoNLL 2003 shared task and for chunking experiments we used the CoNLL 2000 shared task data[12] with standard training, development and testing set splits. The CoNLL '03 and the CoNLL '00 data sets had $\sim 204K/51K/46K$ and $\sim 212K/-/47K$ tokens respectively for Train/Dev./Test sets.

Named Entity Recognition (NER): We use the same set of baseline features as used by (Zhang and Johnson, 2003; Turian et al., 2010) in their experiments. The detailed list of features is as below:

- Current Word w_i; Its type information: all-capitalized, is-capitalized, all-digits and so on; Prefixes and suffixes of w_i

12. More details about the data and competition are available at `http://www.cnts.ua.ac.be/conll2003/ner/` and `http://www.cnts.ua.ac.be/conll2000/chunking/`.

	eigenwords (OSCCA) ($\mu \pm \sigma$)	eigenwords (TSCCA) ($\mu \pm \sigma$)	eigenwords (LR-MVL(I)) ($\mu \pm \sigma$)	eigenwords (LR-MVL(II)) ($\mu \pm \sigma$)	PCA ($\mu \pm \sigma$)	Turian (C&W) ($\mu \pm \sigma$)	Turian (HLBL) ($\mu \pm \sigma$)	SENNA ($\mu \pm \sigma$)	word2vec (SK) ($\mu \pm \sigma$)	word2vec (CB) ($\mu \pm \sigma$)
Positive	25.8± 6.9	24.5± 6.3	26.4± 7.0	29.9± 6.5	33.1 ± 5.8	32.6 ± 6.3	30.0 ± 6.4	29.9 ± 5.1	24.5± 8.3	27.6 ± 7.0
Engagement	18.8 ± 5.0	16.1± 4.4	17.4 ± 4.5	19.6 ± 4.5	29.6 ± 5.2	25.9 ± 5.1	23.2 ± 5.1	20.9 ±5.1	17.2 ± 5.0	20.4 ± 5.3
Relationship	16.3 ± 3.9	**12.2± 3.4**	15.6 ± 4.1	15.9 ± 3.8	46.6 ± 5.4	36.1 ± 5.0	28.3 ± 4.4	18.9 ± 3.4	14.9 ± 4.0	15.0 ± 3.9
Meaningful	10.9 ± 3.9	**8.9± 3.7**	9.5 ± 3.7	9.9 ± 4.0	15.7 ± 3.9	16.1 ± 4.0	15.9 ± 4.0	14.6 ± 3.5	11.1 ± 4.1	14.2 ± 4.5
Achievement	15.7 ± 5.4	**14.6± 5.3**	17.5 ± 5.7	19.0 ± 6.0	30.4 ± 6.0	29.2 ± 6.2	23.0 ± 5.7	20.4 ± 4.9	23.2 ±7.7	27.6 ± 6.8

Table 6: Binary Classification % test errors ($\sum_{i=1}^{n} \frac{\mathbb{I}[y_i \neq \hat{y}_i]}{n}$) averaged over 100 random 80/20 train/test splits for sentiment classification. Bold (3/5 cases) indicates the cases where eigenwords are significantly better (5% level in a paired t-test) compared to all other embeddings. In the remaining 2/5 cases eigenwords are significantly better than all embeddings except word2vec.

- Word tokens in window of 2 around the current word i.e. $d = (w_{i-2}, w_{i-1}, w_i, w_{i+1}, w_{i+2})$; and capitalization pattern in the window.
- Previous two predictions y_{i-1} and y_{i-2} and conjunction of d and y_{i-1}
- Embedding features (eigenwords, C&W, HLBL, Brown etc.) in a window of 2 around the current word including the current word (when applicable).

Following (Ratinov and Roth, 2009) we use a regularized averaged perceptron model with the above set of baseline features for the NER task. We also used their BILOU text chunk representation and fast greedy inference, as it was shown to give superior performance.

We also augment the above set of baseline features with gazetteers, as is standard practice in NER experiments. We also benchmark the performance of eigenwords on MUC7 out-of-domain dataset which had 59K words. MUC7 uses a different annotation and has some different Named Entity types that are not present in the CoNLL '03 dataset, so it provides a good test bed for eigenwords. As earlier, we performed the same preprocessing for this dataset as done by (Turian et al., 2010).

Chunking: For our chunking experiments we use a similar base set of features as above:

- Current Word w_i and word tokens in window of 2 around the current word i.e. $d = (w_{i-2}, w_{i-1}, w_i, w_{i+1}, w_{i+2})$;
- POS tags t_i in a window of 2 around the current word.
- Word conjunction features $w_i \cap w_{i+1}$, $i \in \{-1, 0\}$ and Tag conjunction features $t_i \cap t_{i+1}$, $i \in \{-2, -1, 0, 1\}$ and $t_i \cap t_{i+1} \cap t_{i+2}$, $i \in \{-2, -1, 0\}$.
- Embedding features in a window of 2 around the current word including the current word (when applicable).

Since the CoNLL '00 chunking data does not have a development set, we randomly sampled 1000 sentences from the training data (8936 sentences) for development. So, we trained our chunking models on 7936 training sentences and evaluated their F1 score on the 1000 development sentences and used a CRF[13] as the supervised classifier. We tuned the magnitude of the ℓ_2 regularization penalty in CRF on the development set. The regularization penalty that gave best performance on development set was 2. Finally, we trained the CRF on the entire ("original") training data i.e. 8936 sentences.

9.4.2 Results

The results for NER and chunking are shown in Tables 7 and 8, respectively, which show that eigenwords perform significantly better than state-of-the-art competing methods on both NER and chunking tasks.

9.5 Cross Lingual Word Sense Disambiguation: SEMEVAL 2013

In cross-lingual word sense disambiguation (WSD) tasks, ambiguous English words are given in context as input, and translations of these words into one or more target languages are

13. http://www.chokkan.org/software/crfsuite/

		F1-Score		
Embedding/Model		Dev. Set	Test Set	MUC7
Baseline	No Gazetteers	90.03	84.39	67.48
Brown 1000 clusters		92.32	88.52	78.84
Turian (C&W)		92.46	87.46	75.51
Turian (HLBL)		92.00	88.13	75.25
SENNA		-	88.67	-
word2vec (SK)		92.54	89.40	76.21
word2vec (CB)		92.08	89.20	76.55
eigenwords (OSCCA)		92.94	89.67	79.85
eigenwords (TSCCA)		**93.19**	**89.99**	**80.99**
eigenwords (LR-MVL(I))		92.82	89.85	78.60
eigenwords (LR-MVL(II))		92.73	89.87	78.71
Brown, 1000 clusters	With Gazetteers	93.25	89.41	**82.71**
Turian (C&W)		92.98	88.88	81.44
Turian (HLBL)		92.91	89.35	79.29
SENNA		-	89.59	-
word2vec (SK)		92.99	89.69	79.55
word2vec (CB)		92.93	89.89	79.94
eigenwords (OSCCA)		93.21	90.28	81.59
eigenwords (TSCCA)		**93.96**	**90.59**	82.42
eigenwords (LR-MVL(I))		93.50	90.33	81.15
eigenwords (LR-MVL(II))		93.49	90.10	80.34

Table 7: NER Results. **Note:** F1-score= Harmonic Mean of Precision and Recall. Note that the numbers reported for eigenwords here are different than those in (Dhillon et al., 2011) as we use a different vocabulary size and different dimensionality than there.

Embedding/Model	Test Set F1-Score
Baseline	93.79
Brown 3200 Clusters	94.11
Turian (HLBL)	94.00
Turian (C&W)	94.10
SENNA	93.94
word2vec (SK)	94.02
word2vec (CB)	94.16
eigenwords (OSCCA)	94.02
eigenwords (TSCCA)	**94.23**
eigenwords (LR-MVL(I))	93.97
eigenwords (LR-MVL(II))	94.13

Table 8: Chunking Results. Note that the numbers reported for eigenwords here are different than those in (Dhillon et al., 2011) as we use a different vocabulary size and different dimensionality than there.

produced as output. This can be seen in contrast with more traditional monolingual WSD tasks, in which word senses are instead chosen from a pre-determined sense inventory such as WordNet (Fellbaum, 1998). By framing the problem in a multilingual setting, several important issues are addressed at once. First, by using foreign words rather than human-defined sense labels to resolve ambiguities, WSD systems can more directly be integrated into machine translation and multilingual information retrieval systems, two major areas of application. Moreover, such systems are generalizable to any languages for which sufficient parallel data exists, and do not require the manual construction of sense inventories or sense-tagged corpora for training.

9.5.1 Task Description

We focus on the SemEval 2013 cross-lingual WSD task (Lefever and Hoste, 2013), for which 20 English nouns were chosen for disambiguation. This was framed as an unsupervised task, in which the only provided training data was a sentence-aligned subset of the Europarl parallel corpus (Koehn, 2005). Six languages were included: the source language, English, and the five target languages, namely Spanish, Dutch, German, Italian, and French.

To evaluate a system's output, its answers were compared against the gold standard translations, and corresponding precision and recall scores were computed.

Two evaluation schemes were used in this Semeval task: a Best evaluation metric and an Out-of-Five evaluation metric. For the Best metric, systems could propose multiple sense labels, but the resulting scores were divided by the number of guesses. For the Out-of-Five metric, systems could propose up to five translations without penalty. Further details about this task's evaluation metric can be found in Section 4.1 of Lefever and Hoste (2013).

9.5.2 System Description

Our baseline system was an adaptation of the layer one (L1) classifier described in Section 2 of Rudnick et al. (2013), which was one of the top-scoring systems in the SemEval 2013 cross-lingual WSD task. This system used a maximum entropy model trained on monolingual features from the English source text, incorporating words, lemmas, parts of speech, etc. within a small window of the ambiguous word being classified (Please see Figure 1 of Rudnick et al. (2013) for a detailed list of features). Training instances were extracted programmatically from the provided Europarl subcorpus, using the code made publicly available on the group's GitHub repository[14].

The MEGA Model Optimization Package (MegaM) (Daumé III, 2004) and its NLTK interface (Bird et al., 2009) were used for training the models and producing output for the test sentences.

Using the L1 classifier as a starting point, we began by making two minor modifications to make the system more amenable to further changes. First, regularization was introduced in the form of a Gaussian prior by setting the `sigma` parameter in NLTK's MegaM interface to a nonzero value. Second, "always-on" features were enabled, allowing the classifier to explicitly model the prior probabilities of each output label. Building on this system, we then introduced a variety of embeddings to accompany the existing lexical features. Each

14. https://github.com/hltdi/semeval2013

Best	Spanish	Dutch	German	Italian	French
Most-Frequent Baseline	23.23	20.66	17.43	20.21	25.74
Original L1 System	28.67	21.37	20.64	23.34	27.75
C&W	29.76	**25.17**	22.47	23.59	30.20
HLBL	28.34	24.60	22.35	23.13	29.54
SENNA	**30.78**	24.06	22.39	**25.28**	30.13
Word2Vec (CB)	29.59	25.07	22.73	23.34	30.23
Word2Vec (SK)	29.34	25.04	22.49	23.64	30.09
eigenwords (OSCCA)	30.10	24.58	22.79	24.53	30.37
eigenwords (TSCCA)	30.76	24.56	22.68	24.61	**30.55**
eigenwords (LR-MVL(I))	30.36	24.51	22.92	24.17	30.30
eigenwords (LR-MVL(II))	30.72	24.83	**22.97**	24.85	30.39

Table 9: Best metric F-scores averaged over the twenty English test words.

class of features was included independently of the others in a separate experiment to allow for a direct comparison of the results.

9.5.3 Results

Our experiments were performed using the trial and test data sets from the SemEval 2010 competition, which were released as the trial data for the SemEval 2013 competition. Since the same ambiguous English nouns were tested in both competitions, few changes to the training process were required. The SemEval 2010 trial data was used to select appropriate regularization parameters, and the SemEval 2010 test data was used for the final evaluations.

We used the most frequent translation of an ambiguous word in the training corpus to obtain a baseline score for the Best evaluation metric, and the five most frequent translations to obtain a baseline score for the Out-of-Five evaluation metric. These scores are presented alongside the results of the original L1 classifier and its extensions in Tables 9 and 10. All reported scores are macro averages of the F-scores for the twenty test words from the SemEval 2010 test data. The best score in each category is bolded for emphasis.

We observe that in all cases, the top-scoring system includes some form of vector word embeddings, indicating that these features indeed provide useful information beyond the lexical features from which they are derived. Moreover, the systems using eigenword embeddings outperform the other systems in a majority of cases for both the Best and Out-of-Five evaluation metrics.

9.5.4 Context Specific Embeddings?

The embeddings that we used above for the tasks of NER, Chunking and cross-lingual WSD were the context "oblivious" embeddings i.e. we just used the $\phi_{\mathbf{w}}$ matrix. As described in Section 5 one could induce context specific embeddings also, which help in disambiguating polysemous words. However it turns out that for the tasks of NER, Chunking and WSD they did not give any additional improvement in accuracy. This is due to the fact that in addition to the embedding of the current word we also use the embeddings of words in a window of 2 around the current word as features. They serve as a proxy for the context

Out-of-Five	Spanish	Dutch	German	Italian	French
Most-Frequent Baseline	53.07	43.59	38.86	42.63	51.36
Original L1 System	60.93	46.12	43.40	51.89	57.91
C&W	62.07	48.81	45.06	55.42	63.21
HLBL	61.11	47.25	44.51	55.16	61.19
SENNA	62.88	49.15	45.22	55.92	62.28
Word2Vec (CB)	62.32	48.74	45.51	56.04	62.64
Word2Vec (SK)	61.97	48.35	45.42	56.04	62.55
eigenwords (OSCCA)	62.46	**49.85**	46.34	56.36	62.98
eigenwords (TSCCA)	62.99	49.53	46.60	55.91	63.37
eigenwords (LR-MVL(I))	62.81	49.63	**47.03**	**56.40**	63.12
eigenwords (LR-MVL(II))	**63.05**	49.58	46.86	56.23	**63.51**

Table 10: Out-of-Five metric F-scores averaged over the twenty English test words.

specific embeddings and capture similar discriminative context information as the context specific embeddings do. However, if one only uses the embeddings of the current word as features, then context specific embeddings give improved performance compared to the context oblivious embeddings and the improvement is similar to using the context oblivious embeddings and the embeddings of words in a window of 2 around that word as features.

9.6 Google Semantic and Syntactic Relations Task

(Mikolov et al., 2013a,b) present new syntactic and semantic relation data sets composed of analogous word pairs. The syntactic relations dataset contains word pairs that are different syntactic forms of a given word e.g. write : writes :: eat : eats There are nine such different kinds of relations: adjective-adverb, opposites, comparative, superlative, present participle, nation-nationality, past tense, plural nouns and plural verbs

The semantic relations dataset contains pairs of tuples of word relations that follow a common semantic relation e.g. in Athens : Greece :: Canberra : Australia, where the two given pairs of words follow the country-capital relation. There are three other such kinds of relations: country-currency, man-woman, city-in-state and overall 8869 such pairs of words. The task here is to find a word d that best fits the following relationship: a : b :: c : d given a, b and c. They use the vector offset method, which assumes that the words can be represented as vectors in vector space and computes the offset vector: $y_d = e_a - e_b + e_c$ where e_a, e_b and e_c are the vector embeddings for the words a, b and c. Then, the best estimate of d is the word in the entire vocabulary whose embedding has the highest cosine similarity with y_d. Note that this is a hard problem as it is a v class problem, where v is the vocabulary size.

Table 11 shows the performance of various embeddings for semantic and syntactic relation tasks. Here, as earlier, we trained eigenwords on a Reuters RCV1 with a window size of 2, however as can be seen it performed significantly better compared to all the embeddings except word2vec. We conjectured that it could be due to the fact that we were taking too small a context window which mostly captures syntactic information, which was sufficient for the earlier tasks. So, we experimented with a window size of 10 with the hope that

a broader context window should be able to capture semantic and topic information. For this configuration, the eigenwords' performance was comparable to word2vec and as we had intuited most of the improvement in performance took place on the semantic relation task [15].

Embedding/Model	Semantic Relation	Syntactic Relation	Total Accuracy
Turian (C&W)	1.41	2.20	1.84
Turian (HLBL)	3.33	13.21	8.80
SENNA	9.33	12.35	10.98
eigenwords (Window size= 2) (Best)	12.41	30.27	22.28
word2vec (Window size= 10) (SK)	33.91	32.81	33.30
word2vec (Window size= 10) (CB)	31.05	**36.21**	**33.90**
eigenwords (Window size= 10) (OSCCA)	34.79	31.01	32.70
eigenwords (Window size= 10) (TSCCA)	6.06	10.19	8.34
eigenwords (Window size= 10) (LR-MVL(I))	**35.43**	32.12	33.60
eigenwords (LR-MVL(II))	5.41	19.20	13.03

Table 11: Accuracies for Semantic, Syntactic Relation Tasks and total accuracies.

9.6.1 Which Eigenword Embeddings to Use?

We proposed four algorithms for learning word embeddings and from a practitioners point of view it is natural to ask: Which embedding do I use for my supervised NLP task? Based on the experiments and our experience we found that OSCCA = TSCCA > LR-MVL (I) > LR-MVL(II). In other words, OSCCA and TSCCA work remarkably well out-of-the-box and are robust to the choice of the hidden state dimensionality (k) or the context size (h). Also, since they are not iterative algorithms, they are faster to run than the LR-MVL algorithms. LR-MVL(I) trails the OSCCA and TSCCA algorithms only slightly (not significantly) in terms of performance and sometimes gave better performance than them e.g. on the Google analogy tasks.

The LR-MVL algorithms are different in spirit than OSCCA and TSCCA as they involve an iterative procedure. Unfortunately, since the algorithms involve a CCA operation, they are non-convex and hence there are no convergence guarantees. It might be possible to borrow some theoretical machinery from the alternating minimization literature (Netrapalli et al., 2013) to get convergence bounds, but it is beyond the scope of this paper and we leave it for future work. That said, empirically we never faced any issues regarding multiple local-optima, convergence or matrix inversions. We repeated the process 20 times and both the LR-MVL algorithms gave similar answers.

We found the LR-MVL(II) algorithm to be the least robust and highly sensitive to the values and amounts of smooths used. Its behavior can be explained by its genesis and our motivation for proposing it. LR-MVL(II)) is based on modeling language data using time-series models (in fact exponential smoothing is an ARIMA(0,1,1) process). So, from a modeling perspective LR-MVL(II) has a mature story but still empirically it performs worse than simpler models like OSCCA and TSCCA. This, itself sheds some light on the task of word embedding learning in that simple models work really well and are hard to

15. Note that here TSCCA's performance is significantly worse than other algorithms. This should not be entirely surprising as the theoretical analysis of TSCCA assumes squared loss and those guarantees need not hold after performing vector arithmetic.

beat. Perhaps, its so because the text data is not fully amenable to exponential smoothing, like financial or economic time series data and too small or too big smooths scramble the signal provided by the Zipfian distributed words. Also, since it performs smoothing on one document at a time and is iterative, it can be significantly slower to run.

10. Conclusion & Future Work

In this paper we made two main contributions. First, we proposed four algorithms for learning word embeddings (eigenwords) which are fast to train, have strong theoretical properties, can induce context specific embeddings and have better sample complexity for rare words. All the algorithms had a Canonical Correlation Analysis (CCA) style eigen-decomposition at their core. We performed a thorough evaluation of *eigenwords* learned using these algorithms, and showed that they were comparable to or better than other state-of-the-art algorithms when used as features in a set of NLP classification tasks. Eigenwords are able to capture nuanced syntactic and semantic information about the words. They also have a clearer theoretical foundation than the competing algorithms, which allows us to bound their error rate in recovering the true hidden state under linearity assumptions.

Second, we showed that linear models help us attain state-of-the-art performance on text applications and there is no need to move to more complex non-linear models, e.g. Deep Learning based models. In addition, spectral learning methods are highly scalable and parallelizable and can incorporate the latest advances in numerical linear algebra as black-box routines.

There are many open avenues for future research building on the above spectral methods.

1. Our word embeddings are based on modeling individual words based on their contexts; it will be interesting to induce embeddings for entire phrases or sentences. There are multiple possibilities here. One could directly model phrases by considering a phrase as a "unit" rather than a word, perhaps taking the context of a word or phrase from connected elements in a dependency or constituency parse tree. Another possibility is to learn embeddings for individual words but then combine them in some manner to get an embedding for a phrase or a sentence; some relevant work on this problem has been done by (Socher et al., 2012, 2013).

2. Closely related is the idea of semantic composition. Recent advances in spectral learning for tree structures e.g. (Dhillon et al., 2012a; Cohen et al., 2012) may be able to be extended to provide scalable principled alternative methods to the recursive neural networks of (Socher et al., 2012, 2013).

3. Also it will be fruitful to study embeddings where the contexts are left and right dependencies of a word rather than the neighboring words in the surface structure of the sentence. This might give more precise embeddings with smaller data sets.

4. It will also be interesting to incorporate more domain knowledge into the learning of eigenwords. For example, one could envision using ontologies like WordNet (Fellbaum, 1998) as priors in an otherwise data-driven embedding learning.

Appendix A.

CCA by SVD. Proof of Equation 1:

Proof Assuming $\mathbf{W}$ is the $n \times v$ word matrix and $\mathbf{C}$ is the $n \times hv$ context matrix where n is the number of tokens in the corpus, h is the context size and v is the vocabulary size. Further $\mathbf{C}_{wc} = \mathbf{W}^\top \mathbf{C}$, $\mathbf{C}_{cc} = \mathbf{C}^\top \mathbf{C}$ and $\mathbf{C}_{ww} = \mathbf{W}^\top \mathbf{W}$. The CCA objective is to find vectors ϕ_w and ϕ_c such that the linear combinations $s_w = \phi_w^\top \mathbf{W}$ and $s_{cc} = \phi_c^\top \mathbf{C}$ are maximally correlated i.e.

$$\max_{\phi_w, \phi_c} \frac{{\phi_w}^\top \mathbf{C}_{wc} \phi_c}{\sqrt{{\phi_w}^\top \mathbf{C}_{ww} \phi_w} \sqrt{{\phi_c}^\top \mathbf{C}_{cc} \phi_c}}.$$

This is equivalent to

$$\max_{\phi_w, \phi_c} {\phi_w}^\top \mathbf{C}_{wc} \phi_c,$$

subject to unit-norm constraints ${\phi_w}^\top \mathbf{C}_{ww} \phi_w = I$ and ${\phi_c}^\top \mathbf{C}_{cc} \phi_c = I$.

Then, performing full SVD on $\mathbf{C}_{ww}$ and $\mathbf{C}_{cc}$, we get

$$\begin{aligned} \mathbf{C}_{ww} &= \mathbf{V}_w \mathbf{\Lambda}_w \mathbf{V}_w^\top, \\ \mathbf{C}_{cc} &= \mathbf{V}_c \mathbf{\Lambda}_c \mathbf{V}_c^\top, \end{aligned}$$

where $\mathbf{V}_w^\top \mathbf{V}_w = \mathbf{I}_{v \times v}$ and $\mathbf{V}_c^\top \mathbf{V}_c = \mathbf{I}_{hv \times hv}$.

Define change of basis as

$$\begin{aligned} u_w &= \mathbf{\Lambda}_w^{-1/2} \mathbf{V}_w^\top \mathbf{W}, \\ u_{cc} &= \mathbf{\Lambda}_c^{-1/2} \mathbf{V}_c^\top \mathbf{C}, \end{aligned}$$

Now, in this new transformed basis:

$\mathbb{E}[u_w^\top u_w] = \mathbf{\Lambda}_w^{-1/2} \mathbf{V}_w^\top \mathbf{W} \mathbf{V}_w^\top \mathbf{\Lambda}_w \mathbf{V}_w \mathbf{V}_w \mathbf{\Lambda}_w^{-1/2} = \mathbf{I}_{v \times v}$ and similarly $\mathbb{E}[u_{cc}^\top u_{cc}] = \mathbf{I}_{hv \times hv}$, as desired.

Transform the coefficients ϕ_w and ϕ_c, so that s_w and s_{cc} can be expressed as linear combination in the new basis:

$$\begin{aligned} s_w &= \phi_w^\top \mathbf{W} = g_{\phi_w}^\top u_w \\ s_{cc} &= \phi_c^\top \mathbf{C} = g_{\phi_c}^\top u_{cc} \end{aligned}$$

where $g_{\phi_w} = \mathbf{\Lambda}_w \mathbf{V}_w \phi_w$ and $g_{\phi_c} = \mathbf{\Lambda}_c \mathbf{V}_c \phi_c$.

So, the CCA optimization problem can be cast as the following maximization criteria

$$\max_{g_{\phi_w}, g_{\phi_c}} g_{\phi_w}^\top \mathbf{D}_{wc} g_{\phi_c}$$

subject to unit-norm constraints $g_{\phi_w}^\top g_{\phi_w} = \mathbf{I}$ and $g_{\phi_c}^\top g_{\phi_c} = \mathbf{I}$, where $\mathbf{D}_{wc} = \mathbf{\Lambda}_w^{-1/2} \mathbf{V}_w^\top \mathbf{C}_{wc} \mathbf{V}_c \Lambda_c^{-1/2}$.

The solution to above is nothing but the SVD of $\mathbf{D}_{wc}$.

Finally, we can construct the original coefficient matrices $\boldsymbol{\phi}_w$ and $\boldsymbol{\phi}_c$ as $\boldsymbol{\phi}_w = \mathbf{V}_w \boldsymbol{\Lambda}_w^{-1/2} \mathbf{G}_{\phi_w}$ and $\boldsymbol{\phi}_c = \mathbf{V}_c \boldsymbol{\Lambda}_c^{-1/2} \mathbf{G}_{\phi_c}$, where $\mathbf{G}_{\phi_w}$ and $\mathbf{G}_{\phi_c}$ are the matrices corresponding to the vectors g_{ϕ_w} and g_{ϕ_c} respectively.

Now, in our case $\mathbf{C}_{ww} = \mathbf{W}^\top \mathbf{W}$ is the diagonal word occurrence matrix with the words counts in the corpus on the diagonal, so $\boldsymbol{\Lambda}_w^{-1/2}$ is nothing but $\mathbf{C}_{ww}^{-1/2}$ and $\mathbf{V}_w = \mathbf{I}$.

The context matrix $\mathbf{C}_{cc} = \mathbf{C}^\top \mathbf{C}$, though is not diagonal but it can be approximated by its diagonal. One could also approximate it as a diagonal matrix plus its first order Taylor's expansion, but it would make the resulting matrix substantially more dense and hence the computations intense. In our experiments we found no improvement in prediction accuracy by adding the first order Taylor's term, so we approximate $\mathbf{C}_{cc}$ just by its diagonal. ■

Proof of Theorem 1:

Proof Without loss of generality, we can assume that $\mathbf{W}$ and $\mathbf{C}$ have been transformed to their canonical correlations coordinate space. So $Var(\mathbf{W})$ is the identity and $Var(\mathbf{C})$ is the identity, and the $Cov(\mathbf{W}, \mathbf{C})$ is a diagonal with non-increasing values ρ_i on the diagonal (namely the correlations / singular values). We can write α and β in this coordinate system. By orthogonality we now have $\beta_i = \rho_i \alpha_i$. So, $\mathbb{E}(Y - \beta \mathbf{W})^2$ is simply $\sum (\alpha_i - \beta_i \rho_i)^2$. Which is $\sum \alpha_i^2 (1 - \rho_i^2)$. Our estimator will then have $\gamma_i = \beta_i$ for i smaller than k and $\gamma_i = 0$ otherwise. Hence $(\hat{Y} - \beta^\top \mathbf{W})^2 = \sum_{i=k+1}^{\infty} \beta_i^2$.

So if we pick k to include all terms which have $\rho_i \geq \sqrt{\epsilon}$ our error will be less than $\epsilon \sum_{i=k+1}^{\infty} \alpha_i^2 \leq \epsilon$. ■

Proof of Theorem 2:

Proof The key is that CCA can be understood using the same machinery as is used for analyzing linear regression. In this context we want to recover the word of length v given its context which can be expressed in terms of regression. For a more in-depth discussion of how CCA relates to regression, see (Glahn, 1968), for example. Thus, consider the case of predicting a vector $\mathbf{y}$ of length v (the word) from a vector $\mathbf{x}$ (the context, which is of dimension $2hv$ in the one step CCA case and dimension $2k$ in the two step CCA). We consider the linear model

$$y = \mathbf{x}\beta + \epsilon.$$

Note that, we are predicting only one dimension of our v-dimensional vector $\mathbf{y}$ at a time.

We want to understand the variance of our prediction of a word given the context. As is typical in regression, we calculate a standard error for each coefficient in our contexts, $\approx O(\frac{1}{\sqrt{n}})$. In the one step CCA, $\mathbf{X} = [\mathbf{L} \quad \mathbf{R}]$, and running a regression we will get a prediction error on order of $\frac{hv}{n}$, but since we have v such y's we get a total prediction error on the order of $\frac{hv^2}{n}$.

For the two-step case we take $\mathbf{X} = [\mathbf{L}\phi_{\mathbf{L}} \quad \mathbf{R}\phi_{\mathbf{R}}]$. As mentioned earlier, note that now we are working with about 2k predictors instead of 2hv predictors. If we knew the true $\phi_{\mathbf{L}}$ and $\phi_{\mathbf{R}}$, and thus the true subspace covered by our predictors, the regression error would be on the order of $\frac{kv}{n}$ (again, since there are v entries in our vector $\mathbf{y}$). Instead, we have an estimation of $\phi_{\mathbf{L}}$ and $\phi_{\mathbf{R}}$. If these were computed on infinite amounts of data (and hence

we would be arbitrarily close to the true subspace)–we would be done. However since they come from a sample, we are using $\widehat{\phi_{\mathbf{L}}}$ and $\widehat{\phi_{\mathbf{R}}}$ which are approximation to the ideal $\phi_{\mathbf{L}}$ and $\phi_{\mathbf{R}}$. So our task is to understand the error introduced by this sample approximation of the true CCA. First, we develop some notation and concepts found in (Stewart, 1990).

Consider two subspaces $\mathcal{V}$ and $\hat{\mathcal{V}}$ and respective matrices containing an orthonormal basis for these subspaces $\mathbf{V}$ and $\hat{\mathbf{V}}$. Let $\gamma_1, \gamma_2, \ldots$ be the singular values of the matrix $\mathbf{V}^\top \hat{\mathbf{V}}$, then define

$$\theta_i = \cos^{-1} \gamma_i,$$

and define the canonical angle matrix $\mathbf{\Theta} = \text{diag}(\theta_\mathbf{1}, \ldots, \theta_\mathbf{k})$.

These values of $\mathbf{\Theta}$ capture the effect of using estimated singular vectors, $\hat{\mathbf{V}}$ to form an underlying subspace, as compared to the true subspace formed by the true singular vectors $\mathbf{V}$ stemming from infinite data. The largest canonical angle captures the largest angle between any two vectors- one from the perturbed subspace and one from the true subspace. The second largest canonical angle captures the second largest angle between any two vectors given that they are orthogonal to the original two, and so on. In this proof we will only make use of the largest canonical angle to provide a loose upper bound on the error stemming from the imperfect estimation of the true subspace.

Now, consider a matrix $\hat{\mathbf{A}} = \mathbf{A} + \mathbf{E}$ and take the thin singular value decomposition of $\mathbf{A}$ and $\hat{\mathbf{A}}$ (and here we take the liberty of applying diag in a block matrix sense)

$$\mathbf{A} = [\mathbf{U_1 U_2}]\text{diag}(\mathbf{\Lambda_1}, \mathbf{\Lambda_2})[\mathbf{V_1 V_2}]^\top$$
$$\hat{\mathbf{A}} = [\hat{\mathbf{U}}_\mathbf{1}\hat{\mathbf{U}}_\mathbf{2}]\text{diag}(\hat{\mathbf{\Lambda}}_\mathbf{1}, \hat{\mathbf{\Lambda}}_\mathbf{2})[\hat{\mathbf{V}}_\mathbf{1}\hat{\mathbf{V}}_\mathbf{2}]^\top$$

In our case we have that $\lambda_i = 0$ for all $\lambda_i \in \mathbf{\Lambda_2}$.

From (Stewart and Sun, 1990), we have that

$$\max\{||\sin \mathbf{\Theta}||_\mathbf{2}, ||\sin \mathbf{\Psi}||_\mathbf{2}\} \leq \mathbf{c}||\mathbf{E}||_\mathbf{2}, \tag{3}$$

for some constant c where here $\mathbf{\Theta}$ is the matrix of canonical angles formed from the subspaces of $\mathbf{U}$ and $\hat{\mathbf{U}}$, and $\mathbf{\Psi}$ is the matrix of canonical angles formed between the subspaces of $\mathbf{V}$ and $\hat{\mathbf{V}}$. Note that since $\mathbf{\Theta}$ and $\mathbf{\Psi}$ are diagonal matrices the induced norms $||\cdot||_2$ recover the largest canonical angle of each subspace, and hence we can simultaneously derive an upper bound for the largest canonical angle of either subspace.

We have now developed the machinery we need to analyze the two step CCA.

Without loss of generality, assume that $\mathbf{L}^\top\mathbf{L} = \mathbf{R}^\top\mathbf{R} = \mathbf{I}$ (Even if it is not, we can always rotate $\mathbf{L}$ and $\mathbf{R}$ such that $\mathbf{L}^\top\mathbf{L} = \mathbf{R}^\top\mathbf{R} = \mathbf{I}$ and since PCA/CCA are only identifiable up to a rotation, we would get the same answer.), then ultimately we are interested in projection onto the subspace spanned by $\mathbf{B} = [\mathbf{LU_1} \quad \mathbf{RV_1}]$. Note that because of our assumption the projection onto $\mathbf{LU_1}$ is $\mathbf{LU_1U_1^\top L^\top}$ and similarly for $\mathbf{RV_1}$. Furthermore, note from our assumptions that $\mathbf{LU_1}$ forms an orthonormal basis for the space spanned by $\mathbf{LU_1}$ (since

$$(\mathbf{LU_1})^\top(\mathbf{LU_1}) = \mathbf{U_1^\top L^\top LU_1} = \mathbf{I},$$

and similarly for $\mathbf{L\hat{U}_1}$, $\mathbf{RV_1}$, and $\mathbf{R\hat{V}_1}$).

Lastly, and critically, the singular values of $\mathbf{U_1^\top L^\top L\hat{U}_1}$ are identical to those of $\mathbf{U_1^\top \hat{U}_1}$ (similarly for $\mathbf{RV_1}$ etc.) and so from above we have that the matrix of canonical angles

between the subspaces $\mathbf{LU_1}$ and $\mathbf{L\hat{U}_1}$ are identical to $\mathbf{\Theta}$, the matrix of canonical angles between $\mathbf{U_1}$ and $\mathbf{\hat{U}_1}$, and likewise the matrix of canonical angles between the subspaces $\mathbf{RV_1}$ and $\mathbf{R\hat{V}_1}$ are identical to $\mathbf{\Psi}$, the matrix of canonical angles between $\mathbf{V_1}$ and $\mathbf{\hat{V}_1}$, and thus the maximal angle enjoys the same bound derived above. If we can get a handle on the spectral norm of $\mathbf{E}$, which will come directly from random matrix theory, then we can bound the largest canonical angle of our two subspaces.

We know that $\mathbf{E}$ is a random matrix of iid Gaussian entries with variance $\frac{1}{n}$, and that the largest singular value of a matrix is the spectral norm of the matrix. From random matrix theory we know that the square of the spectral norm of $\mathbf{E}$ is $O(\frac{\sqrt{hv}}{\sqrt{n}})$, from say (Rudelson and Vershynin, 2010).

The strategy will be to divide the variance in the prediction of $\mathbf{y}$ into two separate parts. First the variance that comes from predicting using the incorrect subspace, and then the variance from regression (as stated above) if we had the correct subspace.

Let $\hat{\mathbf{X}} = [\mathbf{L}\hat{\phi}_{\mathbf{L}} \quad \mathbf{R}\hat{\phi}_{\mathbf{R}}]$ (i.e. the incorrect subspace) and $\mathbf{X} = [\mathbf{L}\phi_{\mathbf{L}} \quad \mathbf{R}\phi_{\mathbf{R}}]$ (the true version). To get a handle on predicting with the incorrect subspace (we will consider the subspaces $\mathbf{L}\phi_{\mathbf{L}}$ and $\mathbf{R}\phi_{\mathbf{R}}$ separately here, but note that from (3) the angles between the subspaces and their respective perturbed subspaces are bounded by a common bound) we note that, for the regression of $\mathbf{Y}$ on $\mathbf{X}$ we have

$$\beta|\hat{\mathbf{X}} = \frac{\mathrm{Cov}(\mathbf{Y}, \hat{\mathbf{X}})}{\mathrm{Var}(\hat{\mathbf{X}})},$$

and

$$\beta|\mathbf{X} = \frac{\mathrm{Cov}(\mathbf{Y}, \mathbf{X})}{\mathrm{Var}(\mathbf{X})},$$

and

$$\mathrm{Cov}(\mathbf{Y}, \mathbf{X}) = \mathrm{Cov}(\mathbf{Y}, \hat{\mathbf{X}}),$$

so trivially,

$$\begin{aligned}\beta|\hat{\mathbf{X}} &= \beta|\mathbf{X} * \frac{\mathrm{Var}(\mathbf{X})}{\mathrm{Var}\hat{\mathbf{X}}} \\ &= \beta|\mathbf{X} * \frac{\mathrm{Var}(\mathbf{X})}{\mathrm{Var}(\mathbf{X}) + \mathrm{Var}(\mathbf{X} - \hat{\mathbf{X}})}.\end{aligned}$$

Let $\hat{y}$ be the the estimate of y from the true subspace, and $\hat{\hat{y}}$ be the estimate from the perturbed subspace. For the first part of our strategy, bounding the error that comes from predicting with the incorrect subspace, we want to bound $\mathbb{E}(\hat{y} - \hat{\hat{y}})^2$.

We have,

$$\begin{aligned}
\left[\hat{y} - \hat{\hat{y}}\right]^2 &= \left[\beta|\mathbf{X} * \mathbf{x} - \beta|\hat{\mathbf{X}} * \mathbf{x}\right]^2, \\
&= \left[(\beta|\mathbf{X} - \beta|\hat{\mathbf{X}}) * \mathbf{x}\right]^2, \\
&= \left[\left(\beta|\mathbf{X} - \beta|\mathbf{X}\frac{\mathrm{Var}(\mathbf{X})}{\mathrm{Var}(\mathbf{X}) + \mathrm{Var}(\mathbf{X} - \hat{\mathbf{X}})}\right) * \mathbf{x}\right]^2, \\
&= \left[\beta|\mathbf{X}\left(\mathbf{1} - \frac{\mathrm{Var}(\mathbf{X})}{\mathrm{Var}(\mathbf{X}) + \mathrm{Var}(\mathbf{X} - \hat{\mathbf{X}})}\right) * \mathbf{x}\right]^2, \\
&= \left[\beta|\mathbf{X} * \mathbf{x}\left(\frac{\mathrm{Var}(\mathbf{X} - \hat{\mathbf{X}})}{\mathrm{Var}(\mathbf{X}) + \mathrm{Var}(\mathbf{X} - \hat{\mathbf{X}})}\right)\right]^2, \\
&= \left[\hat{y} * \left(\frac{\mathrm{Var}(\mathbf{X} - \hat{\mathbf{X}})}{\mathrm{Var}(\mathbf{X}) + \mathrm{Var}(\mathbf{X} - \hat{\mathbf{X}})}\right)\right]^2. \qquad (4)
\end{aligned}$$

Because we are working with a ratio of variances instead of actual variances, then without loss of generality we can set $\mathrm{Var}(\hat{\mathbf{X}}) = 1$ for all predictors.

Now, we don't really care what the exact 'true' $\mathbf{X}$'s are (formed with the true singular vectors), because we only care about predicting y and not actually recovering the true β's associated with our SVD. This means we do not suffer from the usual constraints imposed on the erratic behavior of singular vectors. Usually one must handle this kind of error with respect to the entire subspace since singular vectors are highly unstable. In our case, however, we are free to compare to any 'true' vectors we like from the correct subspace, as long as they span the entire true subspace (and nothing more).

We will define a theoretical set of predictors to compare with, then. We are doing this to obtain an upper bound for the total possible variance of $\mathrm{Var}(x - \hat{x})$ for any acceptable set of x's in the true underlying subspace (where we take acceptable to mean that the x's span the true subspace and nothing more).

We handle each subspace $\mathbf{L}\hat{\mathbf{U}}_\mathbf{1}$ and $\mathbf{R}\hat{\mathbf{V}}_\mathbf{1}$ separately. The construction is to take our first vector and choose a vector from the true subspace that lies such that the angle between the two vectors is the maximal canonical angle between the true and perturbed subspaces.

We proceed to our second predictor and choose a vector from the true subspace such the second 'true' predictor is orthogonal to the first. Note that the angle between our second observed $\hat{x}$ and the second chosen x is at most the maximal canonical angle by assumption. Again, because we don't care about the β's associated with our true singular vectors, but only about prediction quality of our perturbed subspace, we need not be worried that our chosen vectors might not be the true singular vectors. We continue in this manner until we have expired all of our predictors from both sets of spaces.

We know from above that the sine of the maximal angle of of both sets of subspaces is $O\left(\frac{\sqrt{hv}}{\sqrt{n}}\right)$ and so we have that the maximal variation

$$\frac{\mathrm{Var}(\mathbf{X} - \hat{\mathbf{X}})}{\mathrm{Var}(\hat{\mathbf{X}})} \sim O\left(\frac{\sqrt{hv}}{\sqrt{n}}\right),$$

and so from 4 we have

$$\mathbb{E}(\hat{y}-\hat{\hat{y}})^2 = \mathbb{E}\left[\hat{y} * O\left(\frac{\sqrt{hv}}{\sqrt{n}}\right)\right]^2$$
$$\approx O\left(\frac{hv}{n} * \frac{1}{v}\right) = O\left(\frac{h}{n}\right).$$

We have v of these to predict, so we have a total error attributable to subspace estimation on the order of $\frac{hv}{n}$. Adding regression error as we did from above, which is on the order of $\frac{kv}{n}$ we get a total error of $\frac{(h+k)v}{n}$. We recall that the error from the one step CCA is on the order of $\frac{hv^2}{n}$ which yields an error ratio of $\frac{h+k}{hv}$. ■

Proof of Lemma 3 and Theorem 4:
Proof Our goal is to find a $v \times k$ matrix $\boldsymbol{\phi}_w$ that maps each of the v words in the vocabulary to a k-dimensional state vector. We will show that the $\boldsymbol{\phi}_w$ we find preserves the information in our data and allows a significant data reduction.

Let $\mathbf{L}$ be an $n \times hv$ matrix giving the words in the left context of each of the n tokens, where the context is of length h, $\mathbf{R}$ be the corresponding $n \times hv$ matrix for the right context, and $\mathbf{W}$ be an $n \times v$ matrix of indicator functions for the words themselves. Also, let l, r and w be the underlying multivariate random variables from which the "observed" matrices $\mathbf{L}$, $\mathbf{R}$ and $\mathbf{W}$ were generated by the data generating process.

We will use three assumptions at various points in our proof:

Assumption 1 *l, r and w come from a rank k HMM i.e it has a rank k observation matrix and a rank k transition matrix both of which have the same domain.*

For example, if the dimension of the hidden state is k and the vocabulary size is v then the observation matrix, which is $k \times v$, has rank k. This rank condition is similar to the one used by (Siddiqi et al., 2010).

Assumption 1A 1 *For the three views, l, r and w assume that there exists a k dimensional "hidden state $\hbar$", such that $\mathbb{E}(l|\hbar) = \hbar\boldsymbol{\beta}_l^\top$ and $\mathbb{E}(r|\hbar) = \hbar\boldsymbol{\beta}_r^\top$ and $\mathbb{E}(w|\hbar) = \hbar\boldsymbol{\beta}_w^\top$ where all $\boldsymbol{\beta}$'s are of rank k.*

This assumption actually follows from the previous one.

Assumption 2 *$\rho(l, w)$, $\rho(l, r)$ and $\rho(w, r)$ all have rank k, where $\rho(a, b)$ is the expected correlation between the random vectors a and b.*

This is a rank condition similar to that in (Hsu et al., 2009).

Assumption 3 *$\rho([l \quad r], w)$ has k distinct singular values.*

This assumption just makes the proof a little cleaner, since if there are repeated singular values, then the singular vectors are not unique. Without it, we would have to phrase results in terms of subspaces with identical singular values.

We also need to define the CCA function that computes the left and right singular vectors for a pair of matrices:

Definition 1 (CCA) *Compute the CCA between multivariate random vectors z and x. Let $\boldsymbol{\phi}_z$ be a matrix containing the d largest singular vectors for z (sorted from the largest on down) and likewise for x. Define the function $CCA(z,x) \equiv [\boldsymbol{\phi}_z, \boldsymbol{\phi}_x]$. When we want just one of these ϕ's, we will use $CCA(z,x)_{left} = \boldsymbol{\phi}_z$ for the left singular vectors and $CCA(z,x)_{right} = \boldsymbol{\phi}_x$ for the right singular vectors.*

Note that the resulting singular vectors, $[\boldsymbol{\phi}_z, \boldsymbol{\phi}_x]$ can be used to give two redundant estimates, $z\boldsymbol{\phi}_z$ and $x\boldsymbol{\phi}_x$ of the "hidden" state relating z and x, if such a hidden state exists.

Lemma 3 *Define $\boldsymbol{\phi}_w$ by the following right singular vectors:*

$$CCA([l \quad r], w)_{right} \equiv \boldsymbol{\phi}_w.$$

Under assumptions 2, 3 and 1A, such that if $CCA(l,r) \equiv [\boldsymbol{\phi}_l, \boldsymbol{\phi}_r]$ then we have

$$CCA([l\boldsymbol{\phi}_l \quad r\boldsymbol{\phi}_r], w)_{right} = \boldsymbol{\phi}_w.$$

This lemma shows that instead of finding the CCA between the full context and the words, we can take the CCA between the Left and Right contexts, estimate a k dimensional state from them, and take the CCA of that state with the words and get the same result.
Proof:
Proof By Assumption 1A, we see that:

$$\mathbb{E}(l\boldsymbol{\beta}_l|\hbar) = \hbar\boldsymbol{\beta}_l^\top\boldsymbol{\beta}_l,$$

and

$$\mathbb{E}(r\boldsymbol{\beta}_r|\hbar) = \hbar\boldsymbol{\beta}_r^\top\boldsymbol{\beta}_r,$$

Since, again by assumption 1Aboth of the $\boldsymbol{\beta}$ matrices have full rank, $\boldsymbol{\beta}_l^\top\boldsymbol{\beta}_l$ is a $k \times k$ matrix of rank k, and likewise for $\boldsymbol{\beta}_r^\top\boldsymbol{\beta}_r$. So

$$\mathbb{E}(\boldsymbol{\beta}_r^\top r^\top l\boldsymbol{\beta}_l|\hbar) = \boldsymbol{\beta}_r^\top\boldsymbol{\beta}_r\hbar^\top\hbar\boldsymbol{\beta}_L\boldsymbol{\beta}_r^\top,$$

i.e.,

$$\boldsymbol{\beta}_r^\top\mathbb{E}(r^\top l)\boldsymbol{\beta}_l = \boldsymbol{\beta}_r^\top\boldsymbol{\beta}_r\mathbb{E}(\hbar^\top\hbar)\boldsymbol{\beta}_l\boldsymbol{\beta}_l^\top,$$

since $\boldsymbol{\beta}_r^\top\boldsymbol{\beta}_r$, $\mathbb{E}(\hbar^\top\hbar)$ and $\boldsymbol{\beta}_l^\top\boldsymbol{\beta}_l$ are all $k \times k$ full rank matrices, $\boldsymbol{\beta}_r$ and $\boldsymbol{\beta}_l$ span the same subspace as the singular values of the CCA between l and r since by Assumption 2 they also have rank k. Similar arguments hold when relating l with w and when relating r with w. Thus if $CCA([l \quad r], w) \equiv [\phi_l, \phi_r]$,

$$CCA(l\boldsymbol{\phi}_l, r\boldsymbol{\phi}_r)_{right} = CCA([l\boldsymbol{\beta}_l \quad r\boldsymbol{\beta}_r], w)_{right},$$

(where we have used Assumption 3 to ensure that not only are the subspaces the same, but that the actual singular vectors are the same.)

Finally by Assumption 3 we know that the rank of $CCA([l \quad r], w)_{right}$ is k, we see that

$$CCA([l\boldsymbol{\beta}_l \quad r\boldsymbol{\beta}_r], w)_{right} = CCA([l \quad r], w)_{right}.$$

Calling this common equality $\boldsymbol{\phi}_w$ yields our result. ■

Let $\boldsymbol{\phi}_w^h$ denote a matrix formed by stacking h copies of $\boldsymbol{\phi}_w$ on top of each other. Right multiplying l or r by $\boldsymbol{\phi}_w^h$ projects each of the words in that context into the k-dimensional reduced rank space.

The following theorem addresses the core of the LR-MVL(II) algorithm, showing that there is an $\boldsymbol{\phi}_w$ which gives the desired dimensionality reduction. Specifically, it shows that the previous lemma also holds in the reduced rank space.

Theorem 4 *Under assumptions 1, 2 and 3 there exists a unique matrix $\boldsymbol{\phi}_w$ such that if*

$$[\boldsymbol{\phi}_l^h, \boldsymbol{\phi}_r^h] \equiv CCA(l\boldsymbol{\phi}_w^h, r\boldsymbol{\phi}_w^h),$$

then

$$\boldsymbol{\phi}_w = CCA([l\boldsymbol{\phi}_w^h\boldsymbol{\phi}_l^h \quad r\boldsymbol{\phi}_w^h\boldsymbol{\phi}_l^h], w)_{right},$$

where $\boldsymbol{\phi}_w^h$ is the stacked form of $\boldsymbol{\phi}_w$.

Proof: We start by noting that Assumption 1 implies Assumption 1A. Thus, the previous lemma follows. So, we know

$$CCA([l \quad r], w)_{right} = CCA([l\boldsymbol{\phi}_l \quad r\boldsymbol{\phi}_r], w)_{right}.$$

Let's define this common quantity as $\boldsymbol{\phi}_w$. This $\boldsymbol{\phi}_w$ has the property that the rank of $CCA(w\boldsymbol{\phi}_w, \hbar)_{left}$ is the same as $CCA(w, \hbar)_{left}$ where $\hbar$ is the hidden state process associated with our data. Hence anything which is not in the domain of $\boldsymbol{\phi}_w$ won't have any correlation with $\hbar$ and hence no correlation with other observed states. So l and $l\boldsymbol{\phi}_w^h$ have the same "information" (predictive power of a linear estimator based on them). More precisely, $[\boldsymbol{\phi}_w^h\boldsymbol{\phi}_l^h, \boldsymbol{\phi}_w^h\boldsymbol{\phi}_r^h] = CCA(l, r)$. Putting this together with the first equation gives the desired result. ■

References

S. Afonso, E. Bick, R. Haber, and D. Santos. Floresta sinta(c)tica: a treebank for portuguese. In *Proceedings of LREC*, pages 1698–1703, 2002.

R. Ando and T. Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853, 2005.

F. Anscombe. The Transformation of Poisson, Binomial and Negative-Binomial data. *Biometrika*, pages 246–254, 1948.

F. Bach and M. Jordan. A probabilistic interpretation of canonical correlation analysis. In *TR 688, University of California, Berkeley*, 2005.

M. Bansal, K. Gimpel, and K. Livescu. Tailoring continuous word representations for dependency parsing. In *Proceedings of ACL*, 2014.

S. Bird and E. Loper. NLTK: The natural language toolkit. In *Proceedings of the ACL 2004 on Interactive Poster and Demonstration Sessions*, ACLdemo '04, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.

S. Bird, E. Klein, and E. Loper. *Natural Language Processing with Python*. O'Reilly Media, 2009.

A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *In Proceedings of COLT*, pages 92–100, 1998.

P. Brown, P.. deSouza, R. Mercer, V. Della Pietra, and J. Lai. Class-based n-gram models of natural language. *Computational Linguistics*, 18:467–479, December 1992. ISSN 0891-2017.

S. Cohen, K. Stratos, M. Collins, D. Foster, and L. Ungar. Spectral learning of latent-variable pcfgs. In *Proceedings of the ACL: Long Papers-Volume 1*, pages 223–231. Association for Computational Linguistics, 2012.

R. Collobert and J. Weston. A unified architecture for natural language processing: deep neural networks with multitask learning. In *Proceedings of ICML*, pages 160–167, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-205-4.

R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12: 2493–2537, 2011.

H. Daumé III. Notes on CG and LM-BFGS Optimization of Logistic Regression. Paper available at `http://pub.hal3.name#daume04cg-bfgs`, implementation available at `http://hal3.name/megam/`, August 2004.

P. Dhillon, D. Foster, and L. Ungar. Multi-view learning of word embeddings via CCA. In *Proceedings of NIPS*, volume 24, 2011.

P. Dhillon, J. Rodu, M. Collins, D. Foster, and L. Ungar. Spectral dependency parsing with latent variables. In *Proceedings of EMNLP-CoNLL*, 2012a.

P. Dhillon, J. Rodu, D. Foster, and L. Ungar. Two Step CCA: A New Spectral Method for Estimating Vector Models of Words. In *Proceedings of ICML*, 2012b.

S. Dumais, G. Furnas, T. Landauer, S. Deerwester, and R. Harshman. Using latent semantic analysis to improve access to textual information. In *Proceedings of SIGCHI Conference on Human Factors in Computing Systems*, pages 281–285. ACM, 1988.

C. Fellbaum. *WordNet*. Wiley Online Library, 1998.

L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman, and E. Ruppin. Placing search in context: The concept revisited. In *Proceedings of WWW*, pages 406–414. ACM, 2001.

D. Foster, S. Kakade, and T. Zhang. Multi-view dimensionality reduction via canonical correlation analysis. Technical report, Technical Report TR-2008-4, TTI-Chicago, 2008.

H. Glahn. Canonical Correlation and Its Relationship to Discriminant Analysis and Multiple Regression. *Journal of the Atmospheric Sciences*, 25(1):23–31, January 1968.

N. Halko, P-G. Martinsson, and J. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 2011.

D. Hardoon and J. Shawe-Taylor. Sparse cca for bilingual word generation. In *EURO Mini Conference, Continuous Optimization and Knowledge-Based Technologies*, 2008.

D. Hardoon, S. Szedmak, and J. Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural Computation*, 16(12):2639–2664, 2004.

H. Hotelling. Canonical correlation analysis (CCA). *Journal of Educational Psychology*, 1935.

D. Hsu, S. Kakade, and T. Zhang. A spectral algorithm for learning hidden markov models. In *Proceedings of COLT*, 2009.

E. Huang, R. Socher, C. Manning, and A. Ng. Improving Word Representations via Global Context and Multiple Word Prototypes. In *Proceedings of ACL:Long Papers-Volume 1*, pages 873–882. Association for Computational Linguistics, 2012.

F. Huang and A. Yates. Distributional representations for handling sparsity in supervised sequence-labeling. In *Proceedings of ACL*, pages 495–503, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. ISBN 978-1-932432-45-9.

F. Huang, A. Ahuja, D. Downey, Y. Yang, Y. Guo, and A. Yates. Learning representations for weakly supervised natural language processing tasks. *Computational Linguistics*, 2013.

D. Jurafsky and J. Martin. *Speech & Language Processing*. Pearson Education India, 2000.

S. Kakade and D. Foster. Multi-view regression via canonical correlation analysis. In Nader H. Bshouty and Claudio Gentile, editors, *In Proceedings of COLT*, volume 4539 of *Lecture Notes in Computer Science*, pages 82–96. Springer, 2007.

P. Koehn. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Conference Proceedings: the tenth Machine Translation Summit*, pages 79–86, Phuket, Thailand, 2005. AAMT. URL `http://mt-archive.info/MTS-2005-Koehn.pdf`.

T. Koo, X. Carreras, and M. Collins. Simple semi-supervised dependency parsing. In *Proceedings of ACL*, 2008.

M. Kromann. The Danish Dependency Treebank and the Underlying Linguistic Theory. In *Proceedings of LREC*, pages 217–220, 2003.

M. Lamar, Y. Maron, M. Johnson, and E. Bienenstock. Svd and clustering for unsupervised pos tagging. In *Proceedings of ACL Short*, pages 215–219, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

T. Landauer, P. Foltz, and D. Laham. An introduction to latent semantic analysis. In *Discourse processes*, 2008.

E. Lefever and V. Hoste. SemEval-2013 Task 10: Cross-Lingual Word Sense Disambiguation. In *In Proceedings of SemEval 2013*, Atlanta, USA, 2013.

M. Marcus, M. Marcinkiewicz, and B. Santorini. Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, 19:313–330, June 1993. ISSN 0891-2017.

T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. 2013a.

T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of NIPS*, pages 3111–3119, 2013b.

A. Mnih and G. Hinton. Three new graphical models for statistical language modelling. In *Proceedings of ICML*, pages 641–648, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-793-3. doi: http://doi.acm.org/10.1145/1273496.1273577. URL `http://doi.acm.org/10.1145/1273496.1273577`.

P. Netrapalli, P. Jain, and S. Sanghavi. Phase Retrieval using Alternating Minimization. In *Proceedings of NIPS*, pages 2796–2804, 2013.

A. Parikh, S. Cohen, and E. Xing. Spectral unsupervised parsing with additive tree metrics. In *Proceedings of ACL*, 2014.

J. Pennebaker, M. Francis, and R. Booth. Linguistic inquiry and word count: LIWC 2001. *Mahway: Lawrence Erlbaum Associates*, 71:2001, 2001.

F. Pereira, N. Tishby, and L. Lee. Distributional clustering of English words. In *In Proceedings of ACL*, pages 183–190, 1993.

L. Ratinov and D. Roth. Design challenges and misconceptions in named entity recognition. In *Proceedings of CONLL*, pages 147–155, 2009.

T. Rose, M. Stevenson, and M. Whitehead. The reuters corpus volume 1-from yesterday's news to tomorrow's language resources. In *Proceedings of LREC*, volume 2, pages 827–832, 2002.

M. Rudelson and R. Vershynin. Non-asymptotic theory of random matrices: extreme singular values, 2010. URL `http://www.citebase.org/abstract?id=oai:arXiv.org:1003.2990`.

A. Rudnick, C. Liu, and M. Gasser. HLTDI: CL-WSD Using Markov Random Fields for SemEval-2013 Task 10. In *Proceedings of SemEval 2013*, 2013.

M. Seligman. *Flourish: A Visionary New Understanding of Happiness and Well-being*. Free Press, 2011.

S. Siddiqi, B. Boots, and G. J. Gordon. Reduced-rank hidden Markov models. In *Proceedings of AISTATS*, 2010.

K. Simov, P. Osenova, M. Slavcheva, S. Kolkovska, E. Balabanova, D. Doikoff, K. Ivanova, A. Simov, E. Simov, and M. Kouylekov. Building a linguistically interpreted corpus of bulgarian: the bultreebank. In *Proceedings of LREC*, 2002.

N. Smith and J. Eisner. Contrastive estimation: Training log-linear models on unlabeled data. In *Proceedings of ACL*, pages 354–362. Association for Computational Linguistics, 2005.

R. Socher, B. Huval, C. Manning, and A. Ng. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the EMNLP-CoNLL*, pages 1201–1211. Association for Computational Linguistics, 2012.

R. Socher, A. Perelygin, J. Wu, J. Chuang, C. Manning, A. Ng, and C. Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of EMNLP*, pages 1631–1642. Citeseer, 2013.

G. Stewart. Perturbation theory for the singular value decomposition. In *SVD and Signal Processing, II: Algoritms, Analysis and Applications*, pages 99–109. Elsevier, 1990.

G. Stewart and J. Sun. *Matrix perturbation theory*. Computer science and scientific computing. Academic Press, 1990. ISBN 9780126702309. URL `http://books.google.com/books?id=l78PAQAAMAAJ`.

J. Suzuki and H. Isozaki. Semi-supervised sequential labeling and segmentation using gigaword scale unlabeled data. In *Proceedings of ACL*, 2008.

O. Täckström, R. McDonald, and J. Uszkoreit. Cross-lingual word clusters for direct transfer of linguistic structure. In *Proceedings of NAACL-HLT*, pages 477–487. Association for Computational Linguistics, 2012.

S. Teufel. *The structure of scientific articles*. CSLI Publications, 2010.

J. Turian, L. Ratinov, and Y. Bengio. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of ACL*, pages 384–394, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. URL `http://portal.acm.org/citation.cfm?id=1858681.1858721`.

P. Turney and P. Pantel. From frequency to meaning: vector space models of semantics. *Journal of Artificial Intelligence Research*, 37:141–188, 2010.

T. Zhang and D. Johnson. A robust risk minimization based named entity recognition system. In *Proceedings of CONLL*, pages 204–207, 2003.

Journal of Machine Learning Research 16 (2015) 3079-3114 Submitted 1/15; Revised 3/15; Published 12/15

Discrete Reproducing Kernel Hilbert Spaces: Sampling and Distribution of Dirac-masses

Palle Jorgensen PALLE-JORGENSEN@UIOWA.EDU
Department of Mathematics
The University of Iowa
Iowa City, IA 52242-1419, U.S.A.

Feng Tian TIANF@TRINE.EDU
Department of Mathematics, Informatics, and Cybersecurity
Trine University
Angola, IN 46703, U.S.A.

Editor: John Shawe-Taylor

Abstract

We study reproducing kernels, and associated reproducing kernel Hilbert spaces (RKHSs) $\mathscr{H}$ over infinite, discrete and countable sets V. In this setting we analyze in detail the distributions of the corresponding Dirac point-masses of V. Illustrations include certain models from neural networks: An Extreme Learning Machine (ELM) is a neural network-configuration in which a hidden layer of weights are randomly sampled, and where the object is then to compute resulting output. For RKHSs $\mathscr{H}$ of functions defined on a prescribed countable infinite discrete set V, we characterize those which contain the Dirac masses δ_x for all points x in V. Further examples and applications where this question plays an important role are: (i) discrete Brownian motion-Hilbert spaces, i.e., discrete versions of the Cameron-Martin Hilbert space; (ii) energy-Hilbert spaces corresponding to graph-Laplacians where the set V of vertices is then equipped with a resistance metric; and finally (iii) the study of Gaussian free fields.

Keywords: Gaussian reproducing kernel Hilbert spaces, sampling in discrete systems, resistance metric, graph Laplacians, discrete Green's functions

1. Introduction

A reproducing kernel Hilbert space (RKHS) is a Hilbert space $\mathscr{H}$ of functions on a prescribed set, say V, with the property that point-evaluation for functions $f \in \mathscr{H}$ is continuous with respect to the $\mathscr{H}$-norm. They are called kernel spaces, because, for every $x \in V$, the point-evaluation for functions $f \in \mathscr{H}$, $f(x)$ must then be given as a $\mathscr{H}$-inner product of f and a vector k_x, in $\mathscr{H}$; called the kernel.

The RKHSs have been studied extensively since the pioneering papers by Aronszajn (1943; 1948). They further play an important role in the theory of partial differential operators (PDO); for example as Green's functions of second order elliptic PDOs (Nelson, 1957; Haeseler et al., 2014). Other applications include engineering, physics, machine-learning theory (Kulkarni and Harman, 2011; Smale and Zhou, 2009; Cucker and Smale, 2002), stochastic processes (Alpay and Dym, 1993; Alpay et al., 1993; Alpay and Dym, 1992; Alpay et al., 2013, 2014), numerical analysis, and more (Lin and Brown, 2004; Ha Quang et al.,

2010; Zhang et al., 2012; Lata and Paulsen, 2011; Vuletić, 2013; Schramm and Sheffield, 2013; Hedenmalm and Nieminen, 2014; Shawe-Taylor and Cristianini, 2004; Schlkopf and Smola, 2001). But the literature so far has focused on the theory of kernel functions defined on continuous domains, either domains in Euclidean space, or complex domains in one or more variables. For these cases, the Dirac δ_x distributions do not have finite $\mathscr{H}$-norm. But for RKHSs over discrete point distributions, it is reasonable to expect that the Dirac δ_x functions will in fact have finite $\mathscr{H}$-norm.

An illustration from neural networks: An Extreme Learning Machine (ELM) is a neural network configuration in which a hidden layer of weights are randomly sampled (Rasmussen and Williams, 2006), and the object is then to determine analytically resulting output layer weights. Hence ELM may be thought of as an approximation to a network with infinite number of hidden units.

Here we consider the discrete case, i.e., RKHSs of functions defined on a prescribed countable infinite discrete set V. We are concerned with a characterization of those RKHSs $\mathscr{H}$ which contain the Dirac masses δ_x for all points $x \in V$. Of the examples and applications where this question plays an important role, we emphasize three: (i) discrete Brownian motion-Hilbert spaces, i.e., discrete versions of the Cameron-Martin Hilbert space; (ii) energy-Hilbert spaces corresponding to graph-Laplacians; and finally (iii) RKHSs generated by binomial coefficients. We show that the point-masses have finite $\mathscr{H}$-norm in cases (i) and (ii), but not in case (iii).

Our setting is a given positive definite function k on $V \times V$, where V is discrete. We study the corresponding RKHS $\mathscr{H}\,(= \mathscr{H}\,(k))$ in detail. Our main results are Theorems 1, 2, and 3 which give explicit answers to the question of which point-masses from V are in $\mathscr{H}$. Applications include Corollaries 29, 41, 46, 48, 52, and 53.

The paper is organized as follows: Section 2 leads up to our characterization (Theorem 1) of point-masses which have finite $\mathscr{H}$-norm. It is applied in Sections 3 and 4 to a variety of classes of discrete RKHSs. Section 3 deals with samples from Brownian motion, and from the Brownian bridge process, and binomial kernels, and with kernels on sets $V \times V$ which arise as restrictions to sample-points. Section 4 covers the case of infinite network of resistors. By this we mean an infinite graph with assigned resistors on its edges. In this family of examples, the associated RKHSs vary with the assignment of resistors on the edges in G, and are computed explicitly from a resulting energy form. Our result Corollary 46 states that, for the network models, all point-masses have finite energy. Furthermore, we compute the value, and we study V as a metric space w.r.t. the corresponding resistance metric. These results, in turn, have direct implications (Corollaries 48, 52 and 55) for the family of Gaussian free fields associated with our infinite network models.

A positive definite kernel k is said to be *universal* (Steinwart, 2002; Caponnetto et al., 2008) if, every continuous function, on a compact subset of the input space, can be uniformly approximated by sections of the kernel, i.e., by continuous functions in the RKHS. In Theorem 3 we show that for the RKHSs from kernels k_c in electrical network G of resistors, this universality holds. The metric in this case is the resistance metric on the vertices of G, determined by the assignment of a conductance function c on the edges in G.

Infinite vs finite graphs. We study "large weighted graphs" (vertices V, edges E, and weights as functions assigned on the edges E), and our motivation derives from learning where "learning" is understood broadly to include (machine) learning of suitable probability

distribution, i.e., meaning learning from samples of training data. Other applications of an analysis of weighted graphs include statistical mechanics, such as infinite spin models, and large digital networks. It is natural to ask then how one best approaches analysis on "large" systems. We propose an analysis via infinite weighted graphs. This is so even if some of the questions in learning theory may in fact refer to only "large" finite graphs.

One reason for this (among others) is that statistical features in such an analysis are best predicted by consideration of probability spaces corresponding to measures on infinite sample spaces. Moreover the latter are best designed from consideration of infinite weighted graphs, as opposed to their finite counterparts. Examples of statistical features which are relevant even for finite samples is long-range order; i.e., the study of correlations between distant sites (vertices), and related phase-transitions, e.g., sign-flips at distant sites. In designing efficient learning models, it is important to understand the possible occurrence of unexpected long-range correlations; e.g., correlations between distant sites in a finite sample.

A second reason for the use of infinite sample-spaces is their use in designing efficient sampling procedures. The interesting solutions will often occur first as vectors in an infinite-dimensional reproducing-kernel Hilbert space RKHS. Indeed, such RKHSs serve as powerful tools in the solution of a kernel-optimization problems with penalty terms. Once an optimal solution is obtained in infinite dimensions, one may then proceed to study its restrictions to suitably chosen finite subgraphs.

In general when reproducing kernels and their Hilbert spaces are used, one ends up with functions on a suitable set, and so far we feel that the dichotomy discrete vs continuous has not yet received sufficient attention. After all, a choice of sampling points in relevant optimization models based on kernel theory suggests the need for a better understanding of point masses as they are accounted for in the RKHS at hand. In broad outline, this is a leading theme in our paper.

2. Discrete RKHSs

Definition 1 *Let V be a countable and infinite set, and $\mathscr{F}(V)$ the set of all* finite *subsets of V. A function $k : V \times V \to \mathbb{C}$ is said to be* positive definite*, if*

$$\sum_{(x,y)\in F\times F}\sum k(x,y)\,\overline{c_x}c_y \geq 0 \tag{1}$$

holds for all coefficients $\{c_x\}_{x\in F} \subset \mathbb{C}$, and all $F \in \mathscr{F}(V)$.

Definition 2 *Fix a set V, countable infinite.*

1. *For all $x \in V$, set*
$$k_x := k(\cdot, x) : V \to \mathbb{C} \tag{2}$$
as a function on V.

2. *Let $\mathscr{H} := \mathscr{H}(k)$ be the Hilbert-completion of the span $\{k_x : x \in V\}$, with respect to the inner product*
$$\left\langle \sum c_x k_x, \sum d_y k_y \right\rangle_{\mathscr{H}} := \sum\sum \overline{c_x} d_y k(x,y) \tag{3}$$

modulo the subspace of functions of zero $\mathscr{H}$-norm. $\mathscr{H}$ is then a reproducing kernel Hilbert space (HKRS), with the reproducing property:

$$\langle k_x, \varphi \rangle_{\mathscr{H}} = \varphi(x), \ \forall x \in V, \ \forall \varphi \in \mathscr{H}. \tag{4}$$

Note. *The summations in (3) are all finite. Starting with finitely supported summations in (3), the RKHS $\mathscr{H} = \mathscr{H}(k)$ is then obtained by Hilbert space completion. We use physicists' convention, so that the inner product is conjugate linear in the first variable, and linear in the second variable.*

3. *If $F \in \mathscr{F}(V)$, set $\mathscr{H}_F$ = closed span$\{k_x\}_{x \in F} \subset \mathscr{H}$, (closed is automatic if F is finite.) And set*

$$P_F := \text{the orthogonal projection onto } \mathscr{H}_F. \tag{5}$$

4. *For $F \in \mathscr{F}(V)$, set*

$$K_F := (k(x,y))_{(x,y) \in F \times F} \tag{6}$$

as a $\#F \times \#F$ matrix.

Remark 3 *It follows from the above that reproducing kernel Hilbert spaces (RKHS) arise from a given positive definite kernel k, a corresponding pre-Hilbert form; and then a Hilbert-completion. The question arises: "What are the functions in the completion?" Now, before completion, the functions are as specified in Definition 2, but the Hilbert space completions are subtle; they are classical Hilbert spaces of functions, not always transparent from the naked kernel k itself. Examples of classical RKHSs: Hardy spaces or Bergman spaces (for complex domains), Sobolev spaces and Dirichlet spaces (Okoudjou et al., 2013; Strichartz and Teplyaev, 2012; Strichartz, 2010) (for real domains, or for fractals), band-limited L^2 functions (from signal analysis), and Cameron-Martin Hilbert spaces from Gaussian processes (in continuous time domain).*

Our focus here is on discrete analogues of the classical RKHSs from real or complex analysis. These discrete RKHSs in turn are dictated by applications, and their features are quite different from those of their continuous counterparts.

Definition 4 *The RKHS $\mathscr{H} = \mathscr{H}(k)$ is said to have the* discrete mass *property ($\mathscr{H}$ is called a* discrete RKHS*), if $\delta_x \in \mathscr{H}$, for all $x \in V$. Here, $\delta_x(y) = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{if } x \neq y \end{cases}$, i.e., the Dirac mass at $x \in V$.*

Lemma 5 *Let $F \in \mathscr{F}(V)$, $x_1 \in F$. Assume $\delta_{x_1} \in \mathscr{H}$. Then*

$$P_F(\delta_{x_1})(\cdot) = \sum_{y \in F} \left(K_F^{-1} \delta_{x_1}\right)(y) \, k_y(\cdot). \tag{7}$$

Proof Show that

$$\delta_{x_1} - \sum_{y \in F} \left(K_F^{-1} \delta_{x_1}\right)(y) \, k_y(\cdot) \in \mathscr{H}_F^{\perp}. \tag{8}$$

The remaining part follows easily from this.

(The notation $(\mathscr{H}_F)^\perp$ stands for orthogonal complement, also denoted $\mathscr{H} \ominus \mathscr{H}_F = \{\varphi \in \mathscr{H} \mid \langle f, \varphi\rangle_{\mathscr{H}} = 0,\ \forall f \in \mathscr{H}_F\}$.) ■

Lemma 6 *Using Dirac's bra-ket, and ket-bra notation (for rank-one operators), the orthogonal projection onto $\mathscr{H}_F$ is*

$$P_F = \sum_{y\in F} \left|k_y\right\rangle\left\langle k_y^*\right|; \tag{9}$$

where

$$k_x^* := \sum_{y\in F} \left(K_F^{-1}\right)_{yx} k_y \tag{10}$$

is the dual vector to k_x, for all $x \in V$.

Proof Let k_x^* be specified as in (10), then

$$\begin{aligned}
\langle k_x^*, k_z\rangle_{\mathscr{H}} &= \sum_{y\in F} \left\langle \left(K_F^{-1}\right)_{yx} k_y, k_z \right\rangle_{\mathscr{H}} \\
&= \sum_{y\in F} \left(K_F^{-1}\right)_{xy} \langle k_y, k_z\rangle_{\mathscr{H}} \\
&= \sum_{y\in F} \left(K_F^{-1}\right)_{xy} (K_F)_{yz} = \delta_{x,z},
\end{aligned}$$

i.e., k_x^* is the dual vector to k_x, for all $x \in V$.

For $f \in \mathscr{H}$, and $F \in \mathscr{F}(V)$, we have

$$\begin{aligned}
\sum_{y\in F} \left|k_y\right\rangle\left\langle k_y^*\right| f &= \sum_{y\in F} \left\langle k_y^*, f\right\rangle_{\mathscr{H}} k_y \\
&= \sum_{(y,z)\in F\times F}\sum \left(K_F^{-1}\right)_{z,y} \langle k_z, f\rangle_{\mathscr{H}} \\
&= P_F f.
\end{aligned}$$

This yields the orthogonal projection realized as stated in (9).

Now, applying (9) to δ_{x_1}, we get

$$\begin{aligned}
P_F(\delta_{x_1}) &= \sum_{y\in F} \left\langle k_y^*, \delta_{x_1}\right\rangle_{\mathscr{H}} k_y \\
&= \sum_{y\in F} \left(\sum_{z\in F} \left(K_F^{-1}\right)_{yz} \langle k_z, \delta_{x_1}\rangle_{\mathscr{H}}\right) k_y \\
&= \sum_{y\in F} \left(\sum_{z\in F} \left(K_F^{-1}\right)_{yz} \delta_{x_1}(z)\right) k_y \\
&= \sum_{y\in F} \left(K_F^{-1}\delta_{x_1}\right)(y)\, k_y,
\end{aligned}$$

where

$$\left(K_F^{-1}\delta_{x_1}\right)(y) := \sum_{z\in F}\left(K_F^{-1}\right)_{yz}\delta_{x_1}(z).$$

This verifies (7). ■

Remark 7 *Note a slight abuse of notations: We make formally sense of the expressions for $P_F(\delta_x)$ in (7) even in the case when δ_x might not be in $\mathscr{H}$. For all finite F, we showed that $P_F(\delta_x)\in\mathscr{H}$. But for δ_x be in $\mathscr{H}$, we must have the additional boundedness assumption (18) satisfied; see Theorem 1.*

Lemma 8 *Let $F\in\mathscr{F}(V)$, $x_1\in F$, then*

$$\left(K_F^{-1}\delta_{x_1}\right)(x_1) = \left\|P_F\delta_{x_1}\right\|_{\mathscr{H}}^2. \tag{11}$$

Proof Setting $\zeta^{(F)} := K_F^{-1}(\delta_{x_1})$, we have

$$P_F(\delta_{x_1}) = \sum_{y\in F}\zeta^{(F)}(y)\,k_F(\cdot,y)$$

and for all $z\in F$,

$$\begin{aligned}\underbrace{\sum_{z\in F}\zeta^{(F)}(z)\,P_F(\delta_{x_1})(z)}_{\zeta^{(F)}(x_1)} &= \sum_F\sum_F\zeta^{(F)}(z)\,\zeta^{(F)}(y)\,k_F(z,y) \qquad (12)\\ &= \left\|P_F\delta_{x_1}\right\|_{\mathscr{H}}^2.\end{aligned}$$

By Lemma 6, the LHS of (12) is given by

$$\begin{aligned}\left\|P_F\delta_{x_1}\right\|_{\mathscr{H}}^2 &= \left\langle P_F\delta_{x_1},\delta_{x_1}\right\rangle_{\mathscr{H}}\\ &= \sum_{y\in F}\left(K_F^{-1}\delta_{x_1}\right)(y)\left\langle k_y,\delta_{x_1}\right\rangle_{\mathscr{H}}\\ &= \left(K_F^{-1}\delta_{x_1}\right)(x_1) = K_F^{-1}(x_1,x_1).\end{aligned}$$

■

Corollary 9 *If $\delta_{x_1}\in\mathscr{H}$ (see Theorem 1), then*

$$\sup_{F\in\mathscr{F}(V)}\left(K_F^{-1}\delta_{x_1}\right)(x_1) = \left\|\delta_{x_1}\right\|_{\mathscr{H}}^2. \tag{13}$$

The following condition is satisfied in some examples, but not all:

Corollary 10 *$\exists F\in\mathscr{F}(V)$ s.t. $\delta_{x_1}\in\mathscr{H}_F\iff$*

$$K_{F'}^{-1}(\delta_{x_1})(x_1) = K_F^{-1}(\delta_{x_1})(x_1)$$

for all $F'\supset F$.

Corollary 11 (Monotonicity) *If F and F' are in $\mathscr{F}(V)$ and $F \subset F'$, then*

$$\left(K_F^{-1}\delta_{x_1}\right)(x_1) \leq \left(K_{F'}^{-1}\delta_{x_1}\right)(x_1) \tag{14}$$

and

$$\lim_{F \nearrow V} \left(K_F^{-1}\delta_{x_1}\right)(x_1) = \|\delta_{x_1}\|_{\mathscr{H}}^2 . \tag{15}$$

Proof By (11),

$$\left(K_F^{-1}\delta_{x_1}\right)(x_1) = \|P_F\delta_{x_1}\|_{\mathscr{H}}^2 .$$

Since $\mathscr{H}_F \subset \mathscr{H}_{F'}$, we have $P_F P_{F'} = P_F$, so

$$\|P_F\delta_{x_1}\|_{\mathscr{H}}^2 = \|P_F P_{F'}\delta_{x_1}\|_{\mathscr{H}}^2 \leq \|P_{F'}\delta_{x_1}\|_{\mathscr{H}}^2$$

i.e.,

$$\left(K_F^{-1}\delta_{x_1}\right)(x_1) \leq \left(K_{F'}^{-1}\delta_{x_1}\right)(x_1) .$$

So (14) follows; and the limit in (15) is monotone. ∎

Theorem 1 *Given V, $k : V \times V \to \mathbb{R}$ positive definite (p.d.). Let $\mathscr{H} = \mathscr{H}(k)$ be the corresponding RKHS. Assume V is countable and infinite. Then the following three conditions (i)-(iii) are equivalent; $x_1 \in V$ is fixed:*

(i) $\delta_{x_1} \in \mathscr{H}$;

(ii) $\exists C_{x_1} < \infty$ *such that for all* $F \in \mathscr{F}(V)$, *the following estimate holds:*

$$|\xi(x_1)|^2 \leq C_{x_1} \sum_{F\times F} \sum \overline{\xi(x)}\xi(y)\, k(x,y) \tag{16}$$

(iii) For $F \in \mathscr{F}(V)$, *set*

$$K_F = \left(k(x,y)\right)_{(x,y)\in F\times F} \tag{17}$$

as a $\#F \times \#F$ *matrix. Then*

$$\sup_{F\in\mathscr{F}(V)} \left(K_F^{-1}\delta_{x_1}\right)(x_1) < \infty. \tag{18}$$

Proof **(i)⇒(ii)** For $\xi \in l^2(F)$, set

$$h_\xi = \sum_{y\in F} \xi(y)\, k_y(\cdot) \in \mathscr{H}_F.$$

Then $\langle \delta_{x_1}, h_\xi \rangle_{\mathscr{H}} = \xi(x_1)$ for all ξ.

Since $\delta_{x_1} \in \mathscr{H}$, then by Schwarz:

$$\left|\langle \delta_{x_1}, h_\xi \rangle_{\mathscr{H}}\right|^2 \leq \|\delta_{x_1}\|_{\mathscr{H}}^2 \sum_{F\times F} \sum \overline{\xi(x)}\xi(y)\, k(x,y) . \tag{19}$$

But $\langle \delta_{x_1}, k_y \rangle_{\mathscr{H}} = \delta_{x_1,y} = \begin{cases} 1 & y = x_1 \\ 0 & y \neq x_1 \end{cases}$; hence $\langle \delta_{x_1}, h_\xi \rangle_{\mathscr{H}} = \xi(x_1)$, and so (19) implies (16).

(ii)⇒(iii) Recall the matrix

$$K_F := (\langle k_x, k_y \rangle)_{(x,y)\in F\times F}$$

as a linear operator $l^2(F) \to l^2(F)$, where

$$(K_F\varphi)(x) = \sum_{y\in F} K_F(x,y)\,\varphi(y),\ \varphi \in l^2(F). \tag{20}$$

By (16), we have

$$\ker(K_F) \subset \left\{\varphi \in l^2(F) : \varphi(x_1) = 0\right\}. \tag{21}$$

Equivalently,

$$\ker(K_F) \subset \{\delta_{x_1}\}^\perp \tag{22}$$

and so $\delta_{x_1}\Big|_F \in \ker(K_F)^\perp = \operatorname{ran}(K_F)$, and $\exists\ \zeta^{(F)} \in l^2(F)$ s.t.

$$\delta_{x_1}\Big|_F = \underbrace{\sum_{y\in F} \zeta^{(F)}(y)\,k(\cdot,y)}_{=:h_F}. \tag{23}$$

Claim. $P_F(\delta_{x_1}) = h_F$, where P_F = projection onto $\mathscr{H}_F$; see (5) and Lemma 5. (See Figure 1.) Indeed, we only need to prove that $\delta_{x_1} - h_F \in \mathscr{H} \ominus \mathscr{H}_F$, i.e.,

$$\langle \delta_{x_1} - h_F, k_z \rangle_{\mathscr{H}} = 0,\ \forall z \in F. \tag{24}$$

But, by (23),

$$\text{LHS}_{(24)} = \delta_{x_1,z} - \sum_{y\in F} k(z,y)\,\zeta^{(F)}(y) = 0.$$

This proves the claim.

If $F \subset F'$, $F, F' \in \mathscr{F}(V)$, then $\mathscr{H}_F \subset \mathscr{H}_{F'}$, and $P_F P_{F'} = P_F$ by easy facts for projections. Hence

$$\|P_F\delta_{x_1}\|^2_{\mathscr{H}} \leq \|P_{F'}\delta_{x_1}\|^2_{\mathscr{H}},\quad h_F := P_F(\delta_{x_1})$$

and

$$\lim_{F\nearrow V} \|\delta_{x_1} - h_F\|_{\mathscr{H}} = 0.$$

(iii)⇒(i) Follows from Lemma 8 and Corollary 9. ∎

Corollary 12 *The numbers* $\left(\zeta^{(F)}(y)\right)_{y\in F}$ *in (23) satisfies*

$$\zeta^{(F)}(x_1) = \sum_{(y,z)\in F\times F}\sum \zeta^{(F)}(y)\,\zeta^{(F)}(z)\,k(y,z). \tag{25}$$

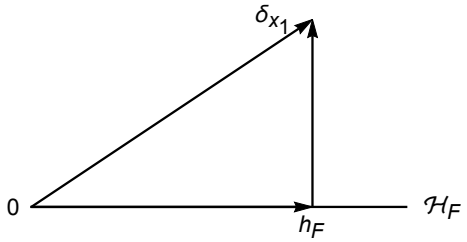


Figure 1: $h_F := P_F(\delta_{x_1})$

Proof Multiply (23) by $\zeta^{(F)}(z)$ and carry out the summation. ■

Remark 13 *To see that (23) is a solution to a linear algebra problem, with $F = \{x_i\}_{i=1}^n$, note that (23) $\Longleftrightarrow$*

$$\begin{bmatrix} k(x_1,x_1) & k(x_1,x_2) & \cdots & k(x_1,x_n) \\ k(x_2,x_1) & k(x_2,x_2) & \cdots & k(x_2,x_n) \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ k(x_n,x_1) & k(x_n,x_2) & \cdots & k(x_n,x_n) \end{bmatrix} \begin{bmatrix} \zeta^{(F)}(x_1) \\ \zeta^{(F)}(x_2) \\ \vdots \\ \zeta^{(F)}(x_{n-1}) \\ \zeta^{(F)}(x_n) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} \tag{26}$$

We now resume the general case of k given and positive definite on $V \times V$.

Corollary 14 *We have*

$$\zeta^{(F)}(x_1) = \|P_F(\delta_{x_1})\|^2_{\mathscr{H}} \tag{27}$$

where

$$P_F(\delta_{x_1}) = \sum_{y\in F} \zeta^{(F)}(y)\, k_y(\cdot) \tag{28}$$

and

$$\zeta^{(F)} = K_N^{-1}(\delta_{x_1}), \quad N := \#F. \tag{29}$$

Proof It follows from (26) that

$$\sum_j k(x_i,x_j)\,\zeta^{(F)}(x_j) = \delta_{1,i}$$

and so multiplying by $\zeta^{(F)}(i)$, and summing over i, gives

$$\underbrace{\sum_i \sum_j k(x_i,x_j)\,\zeta^{(F)}(x_i)\,\zeta^{(F)}(x_j)}_{=\left\|P_F(\delta_{x_1})\right\|^2_{\mathscr{H}}} = \zeta^{(F)}(x_1).$$

■

Corollary 15 *We have*

(i)

$$P_F(\delta_{x_1}) = \zeta^{(F)}(x_1)\, k_{x_1} + \sum_{y \in F \setminus \{x_1\}} \zeta^{(F)}(y)\, k_y \tag{30}$$

where ζ_F solves (26), for all $F \in \mathscr{F}(V)$;

(ii)

$$\|P_F(\delta_{x_1})\|^2_{\mathscr{H}} = \zeta^{(F)}(x_1) \tag{31}$$

and so in particular:

(iii)

$$0 < \zeta^{(F)}(x_1) \leq \|\delta_{x_1}\|^2_{\mathscr{H}} \tag{32}$$

Proof Formula (31) follows from the definition of $\zeta^{(F)}$ as a solution to the matrix problem $K_N \zeta^{(F)} = \delta_{x_1}$, but we may also prove (31) directly from

$$P_F(\delta_{x_1}) = \sum_y \zeta^{(F)}(y)\, k_y\,. \tag{33}$$

Apply $\langle \cdot, \delta_{x_1} \rangle_{\mathscr{H}}$ to both sides in (33), we get

$$\underbrace{\langle \delta_{x_1}, P_F(\delta_{x_1}) \rangle_{\mathscr{H}}}_{\|P_F(\delta_{x_1})\|^2_{\mathscr{H}}} = \zeta^{(F)}(x_1)$$

since $P_F = P_F^* = P_F^2$; i.e., a projection in the RKHS $\mathscr{H} = \mathscr{H}_V$ of k. ∎

Example 1 ($\#F = 2$) *Let $F = \{x_1, x_2\}$, $K_F = (k_{ij})_{i,j=1}^2$, where $k_{ij} := k(x_i, x_j)$. Then (26) reads*

$$\begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix} \begin{bmatrix} \zeta_F(x_1) \\ \zeta_F(x_2) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}. \tag{34}$$

Set $D := \det(K_F) = k_{11}k_{22} - k_{12}k_{21}$, then:

$$\zeta_F(x_1) = \frac{k_{22}}{D}, \quad \zeta_F(x_2) = -\frac{k_{21}}{D}.$$

Example 2 *Let $V = \{x_1, x_2, \ldots\}$ be an ordered set. Set $F_n := \{x_1, \ldots, x_n\}$. Note that with*

$$D_n = \det(K_{F_n}) = \det\left((k(x_i, x_j))_{i,j=1}^n \right), \text{ and} \tag{35}$$

$$D'_{n-1} = (1,1) \text{ minor of } K_{F_n} = \det\left((k(x_i, x_j))_{i,j=2}^n \right); \tag{36}$$

then

$$\zeta^{(F_n)}(x_1) = \frac{D'_{n-1}}{D_n} = \left(K_{F_n}^{-1} \delta_{x_1} \right)(x_1)\,. \tag{37}$$

Corollary 16 *We have*

$$\frac{1}{k(x_1,x_1)} \leq \frac{k(x_2,x_2)}{D_2} \leq \cdots \leq \frac{D'_{n-1}}{D_n} \leq \cdots \leq \|\delta_{x_1}\|^2_{\mathscr{H}}.$$

Proof Follows from (37), and if $F \subset F'$ are two finite subsets, then

$$\|P_F(\delta_{x_1})\|^2_{\mathscr{H}} \leq \|P_{F'}(\delta_{x_1})\|^2_{\mathscr{H}} \leq \|\delta_{x_1}\|^2_{\mathscr{H}}.$$

■

Let $k : V \times V \to \mathbb{R}$ be as specified above. Let $\mathscr{H} = \mathscr{H}(k)$ be the RKHS. We set $\mathscr{F}(V)$:= all finite subsets of V; and if $x \in V$ is fixed, $\mathscr{F}_x(V) := \{F \in \mathscr{F}(V) \mid x \in F\}$.

For $F \in \mathscr{F}(V)$, let K_F be the $\#F \times \#F$ matrix given by $(k(x,y))_{(x,y)\in F\times F}$. Following Karlin and Ziegler (1996), we say that k is *strictly positive* iff $\det K_F > 0$ for all $F \in \mathscr{F}(V)$.

Set $D_F := \det K_F$. If $x \in V$, and $F \in \mathscr{F}_x(V)$, set $K'_F :=$ the minor in K_F obtained by omitting row x and column x, see Figure 2.

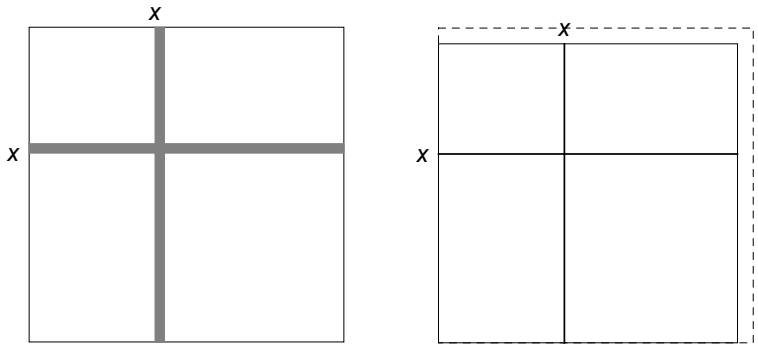


Figure 2: The (x,x) minors, $K_F \to K'_F$.

Corollary 17 *Suppose $k : V \times V \to \mathbb{R}$ is strictly positive. Let $x \in V$. Then*

$$\delta_x \in \mathscr{H} \iff \sup_{F\in\mathscr{F}_x(V)} \frac{D'_F}{D_F} < \infty. \tag{38}$$

2.1 Unbounded Containment in RKHSs

Definition 18 *Let $\mathscr{K}$ and $\mathscr{H}$ be two Hilbert spaces. We say that $\mathscr{K}$ is* unboundedly contained *in $\mathscr{H}$ if there is a dense subspace $\mathscr{K}_0 \subset \mathscr{K}$ such that $\mathscr{K}_0 \subset \mathscr{H}$; and the inclusion operator, with $\mathscr{K}_0$ as its dense domain, is* closed, *i.e.,*

$$\mathscr{K} \overset{incl}{\hookrightarrow} \mathscr{H}, \quad dom(incl) = \mathscr{K}_0.$$

Let $k : V \times V \to \mathbb{R}$ be a p.d. kernel, and let $\mathscr{H}$ be the corresponding RKHS. Set $\mathscr{K} = l^2(V)$, and

$$\mathscr{K}_0 = span\{\delta_x \mid x \in V\}. \tag{39}$$

Proposition 19 *If $\delta_x \in \mathscr{H}$ for $\forall x \in V$, then $l^2(V)$ is unboundedly contained in $\mathscr{H}$.*

Proof Recall that $\mathscr{H}$ is the RKHS defined for a fixed p.d. kernel $k : V \times V \to \mathbb{R}$. Let k_x be the vector in $\mathscr{H}$, given by $k_x(y) = k(x, y)$, s.t.

$$f(x) = \langle k_x, f \rangle_{\mathscr{H}}, \quad \forall f \in \mathscr{H}. \tag{40}$$

To finish the proof we will need: ■

Lemma 20 *The following equation*

$$\langle \delta_x, k_y \rangle_{\mathscr{H}} = \delta_{x,y} \tag{41}$$

holds if $\delta_x \in \mathscr{H}$ for $\forall x \in V$.

Proof (41) is immediate from (40). ■

Lemma 21 *On*

$$span\{k_x \mid x \in V\} \subset \mathscr{H} \tag{42}$$

define $Mk_x := \delta_x$, then by Lemma 20, M extends to be a well defined operator $M : \mathscr{H} \to l^2(V)$ with dense domain (42). We have

$$\langle k, Mf \rangle_{l^2(V)} = \langle k, f \rangle_{\mathscr{H}}, \quad \forall k \in span\{\delta_x\}, \ \forall f \in dom(M). \tag{43}$$

Proof By linearity, it is enough to prove that

$$\langle \delta_x, \delta_y \rangle_{l^2} = \langle \delta_x, k_y \rangle_{\mathscr{H}} \tag{44}$$

holds for $\forall x, y \in V$. But (44) follows immediate from Lemma 20. ■

Corollary 22 *If $L : l^2(V) \to \mathscr{H}$ denotes the inclusion mapping with*

$$dom(L) = span\{\delta_x : x \in V\},$$

then we conclude that

$$L \subset M^*, \text{ and } M \subset L^*. \tag{45}$$

Since $dom(M)$ is dense in $\mathscr{H}$, it follows that L^* has dense domain; and that therefore L is closable.

Remark 23 *This also completes the proof of Proposition 19.*

Corollary 24 *Suppose $k : V \times V \to \mathbb{R}$ is as given, and that $\mathscr{H} = RKHS(k)$. Let L be the densely defined inclusion mapping $l^2(V) \to \mathscr{H}$. Then L^*L is selfadjoint with dense domain in $l^2(V)$; and LL^* is selfadjoint with dense domain in $\mathscr{H}$. Moreover, the following polar decomposition holds:*

$$L = U(L^*L)^{1/2} = (LL^*)^{1/2}U \tag{46}$$

where U is a partial isometry $l^2(V) \to \mathscr{H}$.

3. Point-masses in Concrete Models

Suppose $V \subset D \subset \mathbb{R}^d$ where V is countable and discrete, but D is open. In this case, we get two kernels: k on $D \times D$, and $k_V := k\big|_{V\times V}$ on $V \times V$ by restriction. If $x \in V$, then $k_x^{(V)}(\cdot) = k(\cdot, x)$ is a function on V, while $k_x(\cdot) = k(\cdot, x)$ is a function on D.

This means that the corresponding RKHSs are different, $\mathscr{H}_V$ vs $\mathscr{H}$, where $\mathscr{H}_V$ = a RKHS of functions on V, and $\mathscr{H}$ = a RKHS of functions on D.

Lemma 25 *$\mathscr{H}_V$ is isometrically contained in $\mathscr{H}$ via $k_x^{(V)} \longmapsto k_x$, $x \in V$.*

Proof If $F \subset V$ is a finite subset, and $\xi = \xi_F$ is a function on F, then

$$\left\|\sum\nolimits_{x\in F} \xi(x)\, k_x^{(V)}\right\|_{\mathscr{H}_V} = \left\|\sum\nolimits_{x\in F} \xi(x)\, k_x\right\|_{\mathscr{H}}.$$

The desired result follows from this. ∎

We are concerned with cases of kernels $k : D \times D \to \mathbb{R}$ with restriction $k_V : V \times V \to \mathbb{R}$, where V is a countable discrete subset of D. Typically, for $x \in V$, we may have (restriction) $\delta_x\big|_V \in \mathscr{H}_V$, but $\delta_x \notin \mathscr{H}$; indeed this happens for the kernel k of standard Brownian motion:

$D = \mathbb{R}_+$;

$V =$ an ordered subset $0 < x_1 < x_2 < \cdots < x_i < x_{i+1} < \cdots$, $V = \{x_i\}_{i=1}^{\infty}$.

In this case, we compute $\mathscr{H}_V$, and we show that $\delta_{x_i}\big|_V \in \mathscr{H}_V$; while for $\mathscr{H}_m$ = the Cameron-Martin Hilbert space, we have $\delta_{x_i} \notin \mathscr{H}_m$.

Also note that δ_{x_1} has a different meaning with reference to $\mathscr{H}_V$ vs $\mathscr{H}_m$. In the first case, it is simply $\delta_{x_1}(y) = \begin{cases} 1 & y = x_1 \\ 0 & y \in V\setminus\{x_1\} \end{cases}$. In the second case, δ_{x_1} is a Schwartz distribution. We shall abuse notation, writing δ_x in both cases.

In the following, we will consider restriction to $V \times V$ of a special continuous p.d. kernel k on $\mathbb{R}_+ \times \mathbb{R}_+$. It is $k(s,t) = s \wedge t = \min(s,t)$. Before we restrict, note that the RKHS of this k is the Cameron-Martin Hilbert space of function f on $\mathbb{R}_+$ with distribution derivative $f' \in L^2(\mathbb{R}_+)$, and

$$\|f\|_{\mathscr{H}}^2 := \int_0^\infty \left|f'(t)\right|^2 dt < \infty. \tag{47}$$

For details, see below.

Remark 26 (Application) *The Hilbert space given by $\|\cdot\|_{\mathscr{H}}^2$ in (47) is called the Cameron-Martin Hilbert space, and, as noted, it is the RKHS of $k : \mathbb{R}_+ \times \mathbb{R}_+ \to \mathbb{R} : k(s,t) := s \wedge t$. Now pick a discrete subset $V \subset \mathbb{R}_+$; then Lemma 25 states that the RKHS of the $V \times V$ restricted kernel, $k^{(V)}$ is isometrically embedded into $\mathscr{H}$, i.e., setting*

$$J^{(V)}\left(k_x^{(V)}\right) = k_x, \quad \forall x \in V; \tag{48}$$

$J^{(V)}$ extends by "closed span" to an isometry $\mathscr{H}_V \xrightarrow{J^{(V)}} \mathscr{H}$. It further follows from the lemma, that the range of $J^{(V)}$ may have infinite co-dimension.

Note that $P_V := J^{(V)}\left(J^{(V)}\right)^*$ is the projection onto the range of $J^{(V)}$. The ortho-complement is as follow:

$$\mathscr{H} \ominus \mathscr{H}_V = \left\{\psi \in \mathscr{H} \;\middle|\; \psi(x) = 0,\ \forall x \in V\right\}. \tag{49}$$

Example 3 *Let k and $k^{(V)}$ be as in (48), and set $V := \pi\mathbb{Z}_+$, i.e., integer multiples of π. Then easy generators of wavelet functions (Bratteli and Jorgensen, 2002) yield non-zero functions ψ on $\mathbb{R}_+$ such that*

$$\psi \in \mathscr{H} \ominus \mathscr{H}_V. \tag{50}$$

More precisely,

$$0 < \int_0^\infty \left|\psi'(t)\right|^2 dt < \infty, \tag{51}$$

where ψ' is the distribution (weak) derivative; and

$$\psi(n\pi) = 0, \quad \forall n \in \mathbb{Z}_+. \tag{52}$$

An explicit solution to (50)-(52) is

$$\psi(t) = \prod_{n=1}^{\infty} \cos\left(\frac{t}{2^n}\right) = \frac{\sin t}{t}, \quad \forall t \in \mathbb{R}. \tag{53}$$

From this, one easily generates an infinite-dimensional set of solutions.

3.1 Brownian Motion

Consider the covariance function of standard Brownian motion B_t, $t \in [0, \infty)$, i.e., a Gaussian process $\{B_t\}$ with mean zero and covariance function

$$\mathbb{E}(B_s B_t) = s \wedge t = \min(s, t). \tag{54}$$

We now show that the restriction of (54) to $V \times V$ for an ordered subset (we fix such a set V):

$$V: \ 0 < x_1 < x_2 < \cdots < x_i < x_{i+1} < \cdots \tag{55}$$

has the discrete mass property (Definition 4).

Set $\mathscr{H}_V = RKHS(k|_{V\times V})$,

$$k_V(x_i, x_j) = x_i \wedge x_j. \tag{56}$$

We consider the set $F_n = \{x_1, x_2, \ldots, x_n\}$ of finite subsets of V, and

$$K_n = k^{(F_n)} = \begin{bmatrix} x_1 & x_1 & x_1 & \cdots & x_1 \\ x_1 & x_2 & x_2 & \cdots & x_2 \\ x_1 & x_2 & x_3 & \cdots & x_3 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1 & x_2 & x_3 & \cdots & x_n \end{bmatrix} = (x_i \wedge x_j)_{i,j=1}^n. \tag{57}$$

We will show that condition (iii) in Theorem 1 holds for k_V. For this, we must compute all the determinants, $D_n = \det(K_F)$ etc. ($n = \#F$), see Corollary 17.

Lemma 27

$$D_n = \det\left((x_i \wedge x_j)_{i,j=1}^n\right) = x_1(x_2 - x_1)(x_3 - x_2)\cdots(x_n - x_{n-1}). \tag{58}$$

Proof Induction. In fact,

$$\begin{bmatrix} x_1 & x_1 & x_1 & \cdots & x_1 \\ x_1 & x_2 & x_2 & \cdots & x_2 \\ x_1 & x_2 & x_3 & \cdots & x_3 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1 & x_2 & x_3 & \cdots & x_n \end{bmatrix} \sim \begin{bmatrix} x_1 & 0 & 0 & \cdots & 0 \\ 0 & x_2 - x_1 & 0 & \cdots & 0 \\ 0 & 0 & x_3 - x_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & x_n - x_{n-1} \end{bmatrix},$$

unitary equivalence in finite dimensions. ■

Lemma 28 *Let*

$$\zeta_{(n)} := K_n^{-1}(\delta_{x_1})(\cdot) \tag{59}$$

be as in (11), so that

$$\|P_{F_n}(\delta_{x_1})\|^2_{\mathscr{H}_V} = \zeta_{(n)}(x_1). \tag{60}$$

Then,

$$\begin{aligned} \zeta_{(1)}(x_1) &= \frac{1}{x_1} \\ \zeta_{(n)}(x_1) &= \frac{x_2}{x_1(x_2 - x_1)}, \quad \textit{for } n = 2, 3, \ldots, \end{aligned}$$

and

$$\|\delta_{x_1}\|^2_{\mathscr{H}_V} = \frac{x_2}{x_1(x_2 - x_1)}.$$

Proof A direct computation shows the $(1,1)$ minor of the matrix K_n^{-1} is

$$D'_{n-1} = \det\left((x_i \wedge x_j)_{i,j=2}^n\right) = x_2(x_3 - x_2)(x_4 - x_3)\cdots(x_n - x_{n-1}) \tag{61}$$

and so

$$\begin{aligned} \zeta_{(1)}(x_1) &= \frac{1}{x_1}, \quad \text{and} \\ \zeta_{(2)}(x_1) &= \frac{x_2}{x_1(x_2 - x_1)} \\ \zeta_{(3)}(x_1) &= \frac{x_2(x_3 - x_2)}{x_1(x_2 - x_1)(x_3 - x_2)} = \frac{x_2}{x_1(x_2 - x_1)} \\ \zeta_{(4)}(x_1) &= \frac{x_2(x_3 - x_2)(x_4 - x_3)}{x_1(x_2 - x_1)(x_3 - x_2)(x_4 - x_3)} = \frac{x_2}{x_1(x_2 - x_1)} \\ &\vdots \end{aligned}$$

The result follows from this, and from Corollary 9. ■

Corollary 29 $P_{F_n}(\delta_{x_1}) = P_{F_2}(\delta_{x_1})$, $\forall n \geq 2$. *Therefore,*

$$\delta_{x_1} \in \mathscr{H}_V^{(F_2)} := span\{k_{x_1}^{(V)}, k_{x_2}^{(V)}\} \tag{62}$$

and

$$\delta_{x_1} = \zeta_{(2)}(x_1)\, k_{x_1}^{(V)} + \zeta_{(2)}(x_2)\, k_{x_2}^{(V)} \tag{63}$$

where

$$\zeta_{(2)}(x_i) = K_2^{-1}(\delta_{x_1})(x_i),\ i = 1, 2.$$

Specifically,

$$\zeta_{(2)}(x_1) = \frac{x_2}{x_1(x_2 - x_1)} \tag{64}$$

$$\zeta_{(2)}(x_2) = \frac{-1}{x_2 - x_1}; \tag{65}$$

and

$$\|\delta_{x_1}\|_{\mathscr{H}_V}^2 = \frac{x_2}{x_1(x_2 - x_1)}. \tag{66}$$

Proof Follows from the lemma. Note that

$$\zeta_n(x_1) = \|P_{F_n}(\delta_{x_1})\|_{\mathscr{H}}^2$$

and $\zeta_{(1)}(x_1) \leq \zeta_{(2)}(x_1) \leq \cdots$, since $F_n = \{x_1, x_2, \ldots, x_n\}$. In particular, $\frac{1}{x_1} \leq \frac{x_2}{x_1(x_2 - x_1)}$, which yields (66). ■

Remark 30 *We showed that $\delta_{x_1} \in \mathscr{H}_V$, $V = \{x_1 < x_2 < \cdots\} \subset \mathbb{R}_+$, with the restriction of $s \wedge t$ = the covariance kernel of Brownian motion.*

The same argument also shows that $\delta_{x_i} \in \mathscr{H}_V$ when $i > 1$. We only need to modify the index notation from the case of the proof for $\delta_{x_1} \in \mathscr{H}_V$. The details are sketched below.

Fix $V = \{x_i\}_{i=1}^{\infty}$, $x_1 < x_2 < \cdots$, then

$$P_{F_n}(\delta_{x_i}) = \begin{cases} 0 & \text{if } n < i - 1 \\ \sum_{s=1}^{n} \left(K_{F_n}^{-1} \delta_{x_i}\right)(x_s)\, k_{x_s} & \text{if } n \geq i \end{cases}$$

and

$$\|P_{F_n}(\delta_{x_i})\|_{\mathscr{H}}^2 = \begin{cases} 0 & \text{if } n < i - 1 \\ \frac{1}{x_i - x_{i-1}} & \text{if } n = i \\ \frac{x_{i+1} - x_{i-1}}{(x_i - x_{i-1})(x_{i+1} - x_i)} & \text{if } n > i \end{cases}$$

Conclusion.

$$\delta_{x_i} \in span\left\{k_{x_{i-1}}^{(V)}, k_{x_i}^{(V)}, k_{x_{i+1}}^{(V)}\right\}, \quad \text{and} \tag{67}$$

$$\|\delta_{x_i}\|_{\mathscr{H}}^2 = \frac{x_{i+1} - x_{i-1}}{(x_i - x_{i-1})(x_{i+1} - x_i)}. \tag{68}$$

Corollary 31 *Let $V \subset \mathbb{R}_+$ be countable. If $x_a \in V$ is an accumulation point (from V), then $\|\delta_a\|_{\mathscr{H}_V} = \infty$.*

Remark 32 *This computation will be revisited in Section 4, in a much wider context.*

Example 4 *An illustration for $0 < x_1 < x_2 < x_3 < x_4$:*

$$\begin{aligned} P_F(\delta_{x_3}) &= \sum_{y\in F} \zeta^{(F)}(y)\, k_y(\cdot) \\ \zeta^{(F)} &= K_F^{-1}\delta_{x_3}\,. \end{aligned}$$

That is,

$$\underbrace{\begin{bmatrix} x_1 & x_1 & x_1 & x_1 \\ x_1 & x_2 & x_2 & x_2 \\ x_1 & x_2 & x_3 & x_3 \\ x_1 & x_2 & x_3 & x_4 \end{bmatrix}}_{(K_F(x_i,x_j))_{i,j=1}^4} \begin{bmatrix} \zeta^{(F)}(x_1) \\ \zeta^{(F)}(x_2) \\ \zeta^{(F)}(x_3) \\ \zeta^{(F)}(x_4) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

and

$$\begin{aligned} \zeta^{(F)}(x_3) &= \frac{x_1(x_2-x_1)(x_4-x_2)}{x_1(x_2-x_1)(x_3-x_2)(x_4-x_3)} \\ &= \frac{x_4-x_2}{(x_3-x_2)(x_4-x_3)} = \|\delta_{x_3}\|^2_{\mathscr{H}}\,. \end{aligned}$$

Example 5 (Sparse sample-points) *Let $V = \{x_i\}_{i=1}^\infty$, where*

$$x_i = \frac{i(i-1)}{2}, \quad i \in \mathbb{N}.$$

It follows that $x_{i+1} - x_i = i$, and so

$$\|\delta_{x_i}\|^2_{\mathscr{H}} = \frac{x_{i+1}-x_i}{(x_i-x_{i-1})(x_{i+1}-x_i)} = \frac{2i-1}{(i-1)\,i} \xrightarrow[i\to\infty]{} 0.$$

We conclude that $\|\delta_{x_i}\|_{\mathscr{H}} \xrightarrow[i\to\infty]{} 0$ if the set $V = \{x_i\}_{i=1}^\infty \subset \mathbb{R}_+$ is sparse.

Now, some general facts:

Lemma 33 *Let $k : V \times V \to \mathbb{C}$ be p.d., and let $\mathscr{H}$ be the corresponding RKHS. If $x_1 \in V$, and if δ_{x_1} has a representation as follows:*

$$\delta_{x_1} = \sum_{y\in V} \zeta^{(x_1)}(y)\, k_y\,, \tag{69}$$

then

$$\|\delta_{x_1}\|^2_{\mathscr{H}} = \zeta^{(x_1)}(x_1)\,. \tag{70}$$

Proof Substitute both sides of (69) into $\langle \delta_{x_1}, \cdot \rangle_{\mathscr{H}}$ where $\langle \cdot, \cdot \rangle_{\mathscr{H}}$ denotes the inner product in $\mathscr{H}$. ■

Example 6 (Application) *Suppose $V = \cup_n F_n$, $F_n \subset F_{n+1}$, where each $F_n \in \mathscr{F}(V)$, then if $x_1 \in F_n$, we have*

$$P_{F_n}(\delta_{x_1}) = \sum_{y \in F_n} \left\langle x_1, K_{F_n}^{-1} y \right\rangle_{l^2} k_y \tag{71}$$

and

$$\|P_{F_n}(\delta_{x_1})\|_{\mathscr{H}}^2 = \left\langle x_1, K_{F_n}^{-1} x_1 \right\rangle_{l^2} = \left(K_{F_n}^{-1} \delta_{x_1}\right)(x_1) \tag{72}$$

and the expression $\|P_{F_n}(\delta_{x_1})\|_{\mathscr{H}}^2$ is monotone in n, i.e.,

$$\|P_{F_n}(\delta_{x_1})\|_{\mathscr{H}}^2 \leq \left\|P_{F_{n+1}}(\delta_{x_1})\right\|_{\mathscr{H}}^2 \leq \cdots \leq \|\delta_{x_1}\|_{\mathscr{H}}^2$$

with

$$\sup_{n \in \mathbb{N}} \|P_{F_n}(\delta_{x_1})\|_{\mathscr{H}}^2 = \lim_{n \to \infty} \|P_{F_n}(\delta_{x_1})\|_{\mathscr{H}}^2 = \|\delta_{x_1}\|_{\mathscr{H}}^2 .$$

Question 34 *Let $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ be positive definite, and let $V \subset \mathbb{R}^d$ be a countable discrete subset, e.g., $V = \mathbb{Z}^d$. When does $k\big|_{V \times V}$ have the discrete mass property?*

Examples of the affirmative, or not, will be discussed below.

3.2 Discrete RKHS from Restrictions

Let $D := [0, \infty)$, and $k : D \times D \to \mathbb{R}$, with

$$k(x, y) = x \wedge y = \min(x, y) .$$

Restrict to $V := \{0\} \cup \mathbb{Z}_+ \subset D$, i.e., consider

$$k^{(V)} = k\big|_{V \times V}.$$

$\mathscr{H}(k)$: Cameron-Martin Hilbert space, consisting of functions $f \in L^2(\mathbb{R})$ s.t.

$$\int_0^\infty \left|f'(x)\right|^2 dx < \infty, \quad f(0) = 0.$$

$\mathscr{H}_V := \mathscr{H}(k_V)$. Note that

$$f \in \mathscr{H}(k_V) \Longleftrightarrow \sum_n |f(n) - f(n+1)|^2 < \infty.$$

Lemma 35 *We have $\delta_n = 2k_n - k_{n+1} - k_{n-1}$.*

Proof Introduce the discrete Laplacian Δ, where

$$(\Delta f)(n) = 2f(n) - f(n-1) - f(n+1),$$

then $\Delta k_n = \delta_n$, and

$$\langle 2k_n - k_{n+1} - k_{n-1}, k_m \rangle_{\mathscr{H}_V} = \langle \delta_n, k_m \rangle_{\mathscr{H}_V} = \delta_{n,m}.$$

■

Remark 36 *The same argument as in the proof of the lemma shows (*mutatis mutandis*) that any ordered discrete countable infinite subset $V \subset [0,\infty)$ yields*

$$\mathscr{H}_V := \mathscr{H}\left(k\big|_{V\times V}\right)$$

as a RKHS which is discrete in that (Definition 4) if $V = \{x_i\}_{i=1}^{\infty}$, $x_i \in \mathbb{R}_+$, then $\delta_{x_i} \in \mathscr{H}_V$, $\forall i \in \mathbb{N}$.

Proof Fix vertices $V = \{x_i\}_{i=1}^{\infty}$,

$$0 < x_1 < x_2 < \cdots < x_i < x_{i+1} < \infty, \quad x_i \to \infty. \tag{73}$$

Assign conductance

$$c_{i,i+1} = c_{i+1,i} = \frac{1}{x_{i+1} - x_i}\left(= \frac{1}{\text{dist}}\right) \tag{74}$$

Let

$$\begin{aligned}(\Delta f)(x_i) &= \left(\frac{1}{x_{i+1}-x_i} + \frac{1}{x_i - x_{i-1}}\right) f(x_i) \\ &\quad - \frac{1}{x_i - x_{i-1}} f(x_{i-1}) - \frac{1}{x_{i+1}-x_i} f(x_{i+1})\end{aligned} \tag{75}$$

Equivalently,

$$(\Delta f)(x_i) = (c_{i,i+1} + c_{i,i-1}) f(x_i) - c_{i,i-1} f(x_{i-1}) - c_{i,i+1} f(x_{i+1}). \tag{76}$$

Remark 37 *The most general graph-Laplacians will be discussed in detail in Section 4 below.*

Then, with (76) we have:

$$\Delta k_{x_i} = \delta_{x_i}$$

where $k(\cdot,\cdot)$ = restriction of $s \wedge t$ from $[0,\infty)\times[0,\infty)$ to $V \times V$; and therefore

$$\delta_{x_i} = (c_{i,i+1} + c_{i,i-1}) k_{x_i} - c_{i,i+1} k_{x_{i+1}} - c_{i,i-1} k_{x_{i-1}} \in \mathscr{H}_V \tag{77}$$

as the right-side in the last equation is a finite sum. Note that now the RKHS is

$$\mathscr{H}_V = \left\{ f : V \to \mathbb{C} \;\middle|\; \sum_{i=1}^{\infty} c_{i,i+1} |f(x_{i+1}) - f(x_i)|^2 < \infty \right\}.$$

■

3.3 Brownian Bridge

Let $D := (0,1) =$ the open interval $0 < t < 1$, and set

$$k_{bridge}(s,t) := s \wedge t - st; \tag{78}$$

then (78) is the covariance function for the Brownian bridge $B_{bri}(t)$, i.e.,

$$B_{bri}(0) = B_{bri}(1) = 0 \tag{79}$$

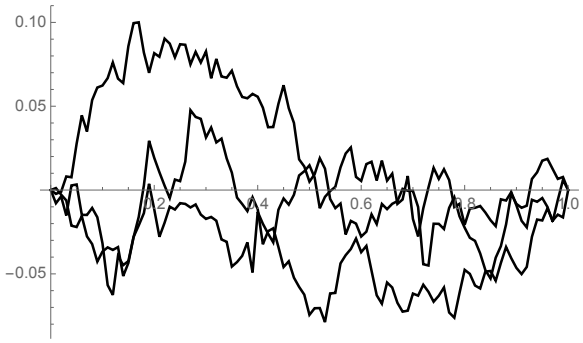


Figure 3: Brownian bridge $B_{bri}(t)$, a simulation of three sample paths of the Brownian bridge.

$$B_{bri}(t) = (1-t)\, B\left(\frac{t}{1-t}\right), \quad 0 < t < 1; \tag{80}$$

where $B(t)$ is Brownian motion; see Lemma 25.

The corresponding Cameron-Martin space is now

$$\mathscr{H}_{bri} = \left\{ f \text{ on } [0,1]\,;\, f' \in L^2(0,1)\,, f(0) = f(1) = 0 \right\} \tag{81}$$

with

$$\|f\|^2_{\mathscr{H}_{bri}} := \int_0^1 \left|f'(s)\right|^2 ds < \infty. \tag{82}$$

If $V = \{x_i\}_{i=1}^{\infty}$, $x_1 < x_2 < \cdots < 1$, is the discrete subset of D, then we have for $F_n \in \mathscr{F}(V)$, $F_n = \{x_1, x_2, \cdots, x_n\}$,

$$K_{F_n} = \left(k_{bridge}(x_i, x_j)\right)_{i,j=1}^n, \tag{83}$$

see (78), and

$$\det K_{F_n} = x_1 (x_2 - x_1) \cdots (x_n - x_{n-1})(1 - x_n). \tag{84}$$

As a result, we get $\delta_{x_i} \in \mathscr{H}_V^{(bri)}$ for all i, and

$$\|\delta_{x_i}\|^2_{\mathscr{H}_V^{(bri)}} = \frac{x_{i+1} - x_{i-1}}{(x_{i+1} - x_i)(x_i - x_{i-1})}.$$

Note $\lim_{x_i \to 1} \|\delta_{x_i}\|^2_{\mathscr{H}_V^{(bri)}} = \infty$.

3.4 Binomial RKHS

Definition 38 *Let $V = \mathbb{Z}_+ \cup \{0\}$; and*

$$k_b(x,y) := \sum_{n=0}^{x\wedge y} \binom{x}{n}\binom{y}{n}, \quad (x,y) \in V \times V.$$

where $\binom{x}{n} = \frac{x(x-1)\cdots(x-n+1)}{n!}$ denotes the standard binomial coefficient from the binomial expansion.

Let $\mathscr{H} = \mathscr{H}(k_b)$ be the corresponding RKHS. Set

$$e_n(x) = \begin{cases} \binom{x}{n} & \text{if } n \le x \\ 0 & \text{if } n > x. \end{cases} \tag{85}$$

Lemma 39 (Alpay and Jorgensen, 2015)

(i) $e_n(\cdot) \in \mathscr{H}$, $n \in V$;

(ii) $\{e_n\}_{n\in V}$ *is an orthonormal basis (ONB) in the Hilbert space* $\mathscr{H}$.

(iii) Set $F_n = \{0,1,2,\ldots,n\}$, *and*

$$P_{F_n} = \sum_{k=0}^{n} |e_k\rangle\langle e_k| \tag{86}$$

or equivalently

$$P_{F_n} f = \sum_{k=0}^{n} \langle e_k, f\rangle_{\mathscr{H}} e_k\,. \tag{87}$$

then,

(iv) Formula (87) is well defined for all functions $f : V \to \mathbb{C}$, $f \in \mathscr{F}unc(V)$.

(v) Given $f \in \mathscr{F}unc(V)$; *then*

$$f \in \mathscr{H} \Longleftrightarrow \sum_{k=0}^{\infty} |\langle e_k, f\rangle_{\mathscr{H}}|^2 < \infty; \tag{88}$$

and, in this case,

$$\|f\|_{\mathscr{H}}^2 = \sum_{k=0}^{\infty} |\langle e_k, f\rangle_{\mathscr{H}}|^2\,.$$

Fix $x_1 \in V$, then we shall apply Lemma 39 to the function $f_1 = \delta_{x_1}$ (in $\mathscr{F}unc(V)$), $f_1(y) = \begin{cases} 1 & \text{if } y = x_1 \\ 0 & \text{if } y \ne x_1. \end{cases}$

Theorem 2 *We have*

$$\left\|P_{F_n}\left(\delta_{x_1}\right)\right\|_{\mathscr{H}}^2 = \sum_{k=x_1}^{n} \binom{k}{x_1}^2.$$

The proof of the theorem will be subdivided in steps; see below.

Lemma 40 (Alpay and Jorgensen, 2015)

(i) For $\forall m, n \in V$, such that $m \leq n$, we have

$$\delta_{m,n} = \sum_{j=m}^{n} (-1)^{m+j} \binom{n}{j}\binom{j}{m}. \tag{89}$$

(ii) For all $n \in \mathbb{Z}_+$, the inverse of the following lower triangle matrix is this: With (see Figure 4)

$$L_{xy}^{(n)} = \begin{cases} \binom{x}{y} & \text{if } y \leq x \leq n \\ 0 & \text{if } x < y \end{cases} \tag{90}$$

we have:

$$\left(L^{(n)}\right)_{xy}^{-1} = \begin{cases} (-1)^{x-y}\binom{x}{y} & \text{if } y \leq x \leq n \\ 0 & \text{if } x < y. \end{cases} \tag{91}$$

Notation: The numbers in (91) are the entries of the matrix $\left(L^{(n)}\right)^{-1}$.

Proof In rough outline, (ii) follows from (i). ∎

$$L^{(n)} = \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & \cdots & 0 & \cdots & 0 & 0 \\ 1 & 1 & 0 & 0 & \cdots & \cdots & 0 & \cdots & 0 & 0 \\ 1 & 2 & 1 & 0 & & & \vdots & & \vdots & \vdots \\ 1 & 3 & 3 & 1 & \ddots & & \vdots & & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & & \vdots & & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & & 1 & 0 & & \vdots & \vdots \\ 1 & \cdots & \binom{x}{y} & \binom{x}{y+1} & \cdots & * & 1 & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & & & & \ddots & 0 & \vdots \\ \vdots & \vdots & \vdots & \vdots & & & & & 1 & 0 \\ 1 & \cdots & \binom{n}{y} & \binom{n}{y+1} & \cdots & \cdots & \cdots & \cdots & n & 1 \end{bmatrix}$$

Figure 4: The matrix L_n is simply a truncated Pascal triangle, arranged to fit into a lower triangular matrix.

Corollary 41 *Let k_b, $\mathscr{H}$, and $n \in \mathbb{Z}_+$ be as above with the lower triangle matrix L_n. Set*

$$K_n(x,y) = k_b(x,y), \quad (x,y) \in F_n \times F_n, \tag{92}$$

i.e., an $(n+1)\times(n+1)$ matrix.

(i) Then K_n is invertible with

$$K_n^{-1} = \left(L_n^{tr}\right)^{-1} (L_n)^{-1}; \tag{93}$$

an (upper triangle) × (lower triangle) factorization.

(ii) For the diagonal entries in the $(n+1)\times(n+1)$ matrix K_n^{-1}, we have:

$$\left\langle x, K_n^{-1} x \right\rangle_{l^2} = \sum_{k=x}^{n} \binom{k}{x}^2$$

Conclusion: Since

$$\left\| P_{F_n}(\delta_{x_1}) \right\|_{\mathscr{H}}^2 = \left\langle x_1, K_n^{-1} x_1 \right\rangle_{\mathscr{H}} \tag{94}$$

for all $x_1 \in F_n$, we get

$$\begin{aligned} \left\| P_{F_n}(\delta_{x_1}) \right\|_{\mathscr{H}}^2 &= \sum_{k=x_1}^{n} \binom{k}{x_1}^2 \\ &= 1 + \binom{x_1+1}{x_1}^2 + \binom{x_1+2}{x_1}^2 + \cdots + \binom{n}{x_1}^2; \end{aligned} \tag{95}$$

and therefore,

$$\left\| \delta_{x_1} \right\|_{\mathscr{H}}^2 = \sum_{k=x_1}^{\infty} \binom{k}{x_1}^2 = \infty.$$

In other words, no δ_x is in $\mathscr{H}$.

4. Infinite Network of Resistors

Here we introduce a family of positive definite kernels $k : V \times V \to \mathbb{R}$, defined on infinite sets V of vertices for a given graph $G = (V, E)$ with edges $E \subset V \times V \backslash$(diagonal).

There is a large literature dealing with analysis on infinite graphs (Jorgensen and Pearse, 2010, 2011, 2013; Okoudjou and Strichartz, 2005; Boyle et al., 2007; Cho and Jorgensen, 2011).

Our main purpose here is to point out that every assignment of resistors on the edges E in G yields a p.d. kernel k, and an associated RKHS $\mathscr{H} = \mathscr{H}(k)$ such that

$$\delta_x \in \mathscr{H}, \quad \text{for all } x \in V. \tag{96}$$

Definition 42 *Let $G = (V, E)$ be as above. Assume*

1. $(x,y) \in E \iff (y,x) \in E$;

2. $\exists c : E \to \mathbb{R}_+$ *(a conductance function = 1 / resistance) such that*

   (i) $c_{(xy)} = c_{(yx)}$, $\forall\, (xy) \in E$;

   (ii) for all $x \in V$, $\#\left\{y \in V \mid c_{(xy)} > 0\right\} < \infty$; *and*

   (iii) $\exists o \in V$ *s.t. for* $\forall x \in V \backslash \{o\}$, $\exists$ *edges* $(x_i, x_{i+1})_0^{n-1} \in E$ *s.t.* $x_o = 0$, *and* $x_n = x$; *called connectedness.*

Given $G = (V, E)$, and a fixed conductance function $c : E \to \mathbb{R}_+$ as specified above, we now define a corresponding Laplace operator $\Delta = \Delta^{(c)}$ acting on functions on V, i.e., on $\mathscr{F}unc\,(V)$ by

$$(\Delta f)\,(x) = \sum_{y \sim x} c_{xy}\,(f\,(x) - f\,(y))\,. \tag{97}$$

Let $\mathscr{H}$ be the Hilbert space defined as follows: A function f on V is in $\mathscr{H}$ iff $f\,(o) = 0$, and

$$\|f\|^2_{\mathscr{H}} := \frac{1}{2} \sum_{\substack{(x,y) \in E \\ \subset V \times V}} \sum c_{xy}\,|f\,(x) - f\,(y)|^2 < \infty. \tag{98}$$

Lemma 43 (Jorgensen and Pearse, 2010) *For all* $x \in V \backslash \{o\}$, $\exists v_x \in \mathscr{H}$ *s.t.*

$$f\,(x) - f\,(o) = \langle v_x, f \rangle_{\mathscr{H}}\,, \quad \forall f \in \mathscr{H} \tag{99}$$

where

$$\langle h, f \rangle_{\mathscr{H}} = \frac{1}{2} \sum_{(x,y) \in E} \sum c_{xy} \left(\overline{h\,(x)} - \overline{h\,(y)}\right)(f\,(x) - f\,(y))\,, \quad \forall h, f \in \mathscr{H}. \tag{100}$$

(The system $\{v_x\}$ *is called a system of dipoles.)*

Proof Let $x \in V \backslash \{o\}$, and use (97) together with the Schwarz-inequality to show that

$$|f\,(x) - f\,(o)|^2 \leq \sum_i \frac{1}{c_{x_i x_{i+1}}} \sum_i c_{x_i x_{i+1}}\,|f\,(x_i) - f\,(x_{i+1})|^2\,.$$

An application of Riesz' lemma then yields the desired conclusion.

Note that $v_x = v_x^{(c)}$ depends on the choice of base point $o \in V$, and on conductance function c; see (i)-(ii) and (98). ∎

Now set

$$k^{(c)}\,(x, y) = \langle v_x, v_y \rangle_{\mathscr{H}}\,, \quad \forall\, (xy) \in (V \backslash \{o\}) \times (V \backslash \{o\})\,. \tag{101}$$

It follows from a theorem that $k^{(c)}$ is a Green's function for the Laplacian $\Delta^{(c)}$ in the sense that

$$\Delta^{(c)} k^{(c)}\,(x, \cdot) = \delta_x \tag{102}$$

where the dot in (102) is the dummy-variable in the action. Note that the solution to (102) is not unique.

Lemma 44 (Jorgensen and Pearse, 2011) *Let $G = (V, E)$, and conductance function $c : E \to \mathbb{R}_+$ be a s specified above; then $k^{(c)}$ in (101) is positive definite, and the corresponding RKHS $\mathscr{H}\left(k^{(c)}\right)$ is the Hilbert space introduced in (98) and (100), called the energy-Hilbert space.*

Proof See Jorgensen et al. (2010; 2011; 2013). ■

Proposition 45 *Let $x \in V \setminus \{o\}$, and let $c : E \to \mathbb{R}_+$ be specified as above. Let $\mathscr{H} = \mathscr{H}(k^c)$ be the corresponding RKHS. Then $\delta_x \in \mathscr{H}$, and*

$$\|\delta_x\|^2_{\mathscr{H}} = \sum_{y \sim x} c_{(xy)} =: c(x). \tag{103}$$

Proof We study the finite matrices, defined for $\forall F \in \mathscr{F}(V)$, by

$$K_F(x, y) = k^c(x, y), \quad (x, y) \in F \times F. \tag{104}$$

Fix $x \in V \setminus \{o\}$, and pick $F \in \mathscr{F}(V)$ such that

$$\{x\} \cup \{y \in V \mid y \sim x\} \subset F, \tag{105}$$

see Figure 5; an interior point:

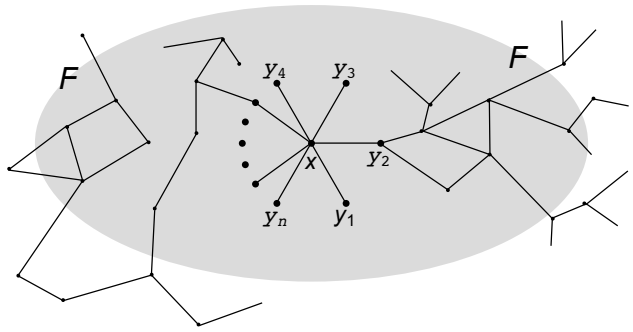


Figure 5: Neighborhood of x, see Definition 42 (ii). An interior point x.

Let $F \in \mathscr{F}(V)$ be as in (104) and in Figure 5, and let $\Delta = \Delta^{(c)}$ be the Laplace operator (97), then for all $(x, y) \in F \times F$, we have:

$$\begin{aligned} \left\langle x, K_F^{-1} y \right\rangle_{l^2} &= \langle \delta_x, \Delta \delta_y \rangle_{l^2} \\ &= (\Delta \delta_y)(x) \\ &= \begin{cases} c(x) & \text{if } y = x; \text{ see (103)} \\ -c_{(xy)} & \text{if } y \sim x \\ 0 & \text{for all other values of } y \end{cases} \end{aligned} \tag{106}$$

In particular,

$$\sup_{F\in\mathscr{F}(V)} \left(K_F\delta_x\right)(x) < \infty;$$

and in fact,

$$\|\delta_x\|^2_{\mathscr{H}} = c\,(x)\,,\ \text{for all } x\in V\setminus\{o\},$$

as claimed in the Proposition.

The last step in the present proof uses the equivalence (i)⇔(ii)⇔(iii) from Theorem 1 above.

Finally, we note that the assertion in (106) follows from

$$\Delta v_x = \delta_x - \delta_o, \quad \forall x\in V\setminus\{o\}\,. \tag{107}$$

And (107) in turn follows from (99), (97) and a straightforward computation. ■

Corollary 46 *Let $G=(V,E)$ and conductance $c: E\to\mathbb{R}_+$ be as specified above. Let $\Delta=\Delta^{(c)}$ be the corresponding Laplace operator. Let $\mathscr{H}=\mathscr{H}\left(k^c\right)$ be the RKHS. Then*

$$\langle\delta_x, f\rangle_{\mathscr{H}} = \left(\Delta f\right)(x) \tag{108}$$

and

$$\delta_x = c\,(x)\,v_x - \sum_{y\sim x} c_{xy}v_y \tag{109}$$

holds for all $x\in V$.

Proof Since the system $\{v_x\}$ of dipoles in (99) span a dense subspace in $\mathscr{H}$, it is enough to verify (108) when $f=v_y$ for $y\in V\setminus\{o\}$. But in this case, (108) follows from (102) and (106). ■

Corollary 47 *Let $G=(V,E)$, and conductance $c: E\to\mathbb{R}_+$ be as before; let $\Delta^{(c)}$ be the Laplace operator, and $\mathscr{H}_E^{(c)}$ the energy-Hilbert space in Definition 42 (Equation (98)). Let $k^{(c)}(x,y)=\langle v_x, v_y\rangle_{\mathscr{H}_E}$ be the kernel from (101), i.e., the Green's function of $\Delta^{(c)}$. Then the two Hilbert spaces $\mathscr{H}_E$, and $\mathscr{H}\left(k^{(c)}\right)=RKHS\left(k^{(c)}\right)$, are naturally isometrically isomorphic via $v_x\longmapsto k_x^{(c)}$ where $k_x^{(c)}=k^{(c)}(x,\cdot)$ for all $x\in V$.*

Proof Let $F\in\mathscr{F}(V)$, and let ξ be a function on F; then

$$\begin{aligned}
\left\|\sum\nolimits_{x\in F}\xi\,(x)\,k_x^{(c)}\right\|^2_{\mathscr{H}\left(k^{(c)}\right)} &= \sum_{F\times F}\sum\overline{\xi\,(x)}\xi\,(y)\,k^{(c)}\,(x,y)\\
&\underset{(101)}{=} \sum_{F\times F}\sum\overline{\xi\,(x)}\xi\,(y)\,\langle v_x, v_y\rangle_{\mathscr{H}_E}\\
&= \left\|\sum\nolimits_{x\in F}\xi\,(x)\,v_x\right\|^2_{\mathscr{H}_E}.
\end{aligned}$$

The remaining steps in the proof of the Corollary now follows from the standard completion from dense subspaces in the respective two Hilbert spaces $\mathscr{H}_E$ and $\mathscr{H}\left(k^{(c)}\right)$. ■

In the following we show how the kernels $k^{(c)} : V \times V \to \mathbb{R}$ from (101) in Lemma 43 are related to metrics on V; so called *resistance metrics* (Jorgensen and Pearse, 2010; Alpay et al., 2013).

Corollary 48 *Let $G = (V, E)$, and conductance $c : E \to \mathbb{R}_+$ be as above; and let $k^{(c)}(x, y) := \langle v_x, v_y \rangle_{\mathscr{H}_E}$ be the corresponding Green's function for the graph Laplacian $\Delta^{(c)}$.*

Then there is a metric $R\left(= R^{(c)} = \text{the resistance metric}\right)$, such that

$$k^{(c)}(x, y) = \frac{R^{(c)}(o, x) + R^{(c)}(o, y) - R^{(c)}(x, y)}{2} \tag{110}$$

holds on $V \times V$. Here the base-point $o \in V$ is chosen and fixed s.t.

$$\langle V_x, f \rangle_{\mathscr{H}_E} = f(x) - f(o), \quad \forall f \in \mathscr{H}_E, \ \forall x \in V. \tag{111}$$

Proof Set

$$R^{(c)}(x, y) = \|v_x - v_y\|^2_{\mathscr{H}_E}. \tag{112}$$

We proved (Jorgensen and Pearse, 2010) that $R^{(c)}(x, y)$ in (112) indeed defines a metric on V; the so called *resistance metric*. It represents the voltage-drop from x to y when 1 Amp is fed into (G, c) at the point x, and then extracted at y.

The verification of (110) is now an easy computation, as follows:

$$\begin{aligned} & \frac{R^{(c)}(o, x) + R^{(c)}(o, y) - R^{(c)}(x, y)}{2} \\ = \ & \frac{\|v_x\|^2_{\mathscr{H}_E} + \|v_y\|^2_{\mathscr{H}_E} - \|v_x - v_y\|^2_{\mathscr{H}_E}}{2} \\ = \ & \langle v_x, v_y \rangle_{\mathscr{H}_E} \\ = \ & k^{(c)}(x, y) \quad \text{by (101).} \end{aligned}$$

■

Proposition 49 *In the two cases: (i) $B(t)$, Brownian motion on $0 < t < \infty$; and (ii) the Brownian bridge $B_{bri}(t)$, $0 < t < 1$, from Section 3 (Figure 3), the corresponding resistance metric R is as follows:*

(i) If $V = \{x_i\}_{i=1}^{\infty} \subset (0, \infty)$, $x_1 < x_2 < \cdots$, then

$$R_B^{(V)}(x_i, x_j) = |x_i - x_j|. \tag{113}$$

(ii) If $W = \{x_i\}_{i=1}^{\infty} \subset (0, 1)$, $0 < x_1 < x_2 < \cdots < 1$, then

$$R_{bridge}^{(W)}(x_i, x_j) = |x_i - x_j| \cdot (1 - |x_i - x_j|). \tag{114}$$

In the completion w.r.t. the resistance metric $R_{bridge}^{(W)}$, the two endpoints $x = 0$ and $x = 1$ are identified.

4.1 Gaussian Processes

Definition 50 *A* Gaussian realization *of an infinite graph-network $G = (V, E)$, with prescribed conductance function $c : E \to \mathbb{R}_+$, and* dipoles $(v_x^c)_{x \in V \setminus \{o\}}$, *is a Gaussian process $(X_x)_{x \in V}$ on a probability space $(\Omega, \mathscr{F}, \mathbb{P})$, where Ω is a sample space; $\mathscr{F}$ a sigma-algebra of events, and $\mathbb{P}$ a probability measure s.t., for $\forall F \in \mathscr{F}(V)$, the random variables $(X_x)_{x \in F}$, are jointly Gaussian with*

$$\mathbb{E}(X_x) = \int_\Omega X_x d\mathbb{P} = 0 \tag{115}$$

and covariance

$$\mathbb{E}(X_x X_y) = k^{(c)}(x, y) = \left\langle v_x^{(c)}, v_y^{(c)} \right\rangle_{\mathscr{H}_E}; \tag{116}$$

i.e., the covariance matrix $(\mathbb{E}(X_x X_y))_{(x,y) \in F \times F}$ is

$$K_F(x, y) := k^{(c)}(x, y) \text{ on } F \times F. \tag{117}$$

Lemma 51 (Jorgensen and Pearse, 2010) *For all $G = (V, E)$, and $c : E \to \mathbb{R}_+$, as specified, Gaussian realizations exist; they are called Gaussian free fields.*

Corollary 52 *Let $G = (V, E)$, $c : E \to \mathbb{R}_+$ be as above; and let $(X_x)_{x \in V}$ be an associated Gaussian free field. Then the point Dirac-masses $(\delta_x)_{x \in V}$ have Gaussian realizations*

$$\widetilde{\delta_x} = c(x) X_x - \sum_{y \sim x} c_{xy} X_y, \quad \forall x \in V. \tag{118}$$

Corollary 53 *Let $G = (V, E)$, and $c : E \to \mathbb{R}_+$ be as above. Let $\{X_x\}_{x \in V}$ be the corresponding Gaussian free field, i.e., with correlation*

$$\mathbb{E}(X_x X_y) = k^{(c)}(x, y) = \left\langle v_x^{(c)}, v_y^{(c)} \right\rangle_{\mathscr{H}_E} \tag{119}$$

where the dipoles $\{v_x^{(c)}\} \subset \mathscr{H}_E$ are computed w.r.t. a chosen (and fixed) based-point $o \in V$, i.e.,

$$\left\langle v_x^{(c)}, f \right\rangle_{\mathscr{H}_E} = f(x) - f(o), \quad \forall f \in \mathscr{H}_E,\ x \in V. \tag{120}$$

Finally, let $R^{(c)}(x, y)$ be the corresponding resistance metric on V. Then

$$\mathbb{E}(X_x X_z) + \mathbb{E}(X_z X_y) \leq \mathbb{E}(X_x X_y) + R^{(c)}(o, z) \tag{121}$$

holds for all vertices $x, y, z \in V$; see Figure 6.

Proof Use Corollary 48, and (112). We have

$$\|v_x - v_y\|_{\mathscr{H}}^2 \leq \|v_x - v_z\|_{\mathscr{H}}^2 + \|v_z - v_y\|_{\mathscr{H}}^2,$$

and (121) now follows from (116). ∎

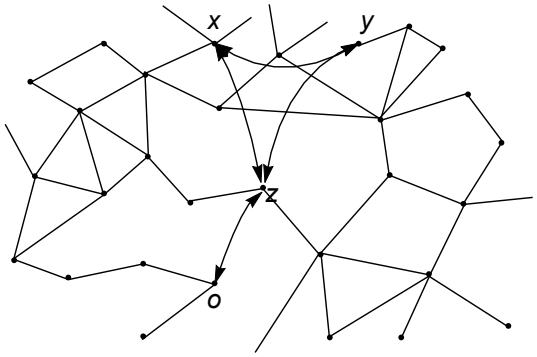


Figure 6: Covariance vs resistance distance $R^{(c)}(o,z)$ for three vertices $x,y,z\in V$.

4.2 Metric Completion

The next theorem illustrates a connection between the universal property of a kernel in a RKHS $\mathscr{H}$, on the one hand, and the distribution of the Dirac point-masses δ_x, on the other. We make "distribution" precise by the quantity $E(x):=\|\delta_x\|^2_{\mathscr{H}}$, the energy of the point-mass at the vertex point x. We introduce a metric completion M, and the universal property of the RKHS $\mathscr{H}$ asserts that the functions from $\mathscr{H}$ are continuous and 1/2-Lipschitz on M, and that they approximate every continuous function on M in the uniform norm. Recall, the vertex set V is equipped with its resistance metric. The universal property here refers to the corresponding metric completion M of the discrete vertex set. In the interesting cases (see e.g., Example 7), M is a continuum; in the case of the example below, the boundary of V is a Cantor set. One expects the value of $E(x)$ to go to infinity as x approaches the boundary M, and this is illustrated in the example; with an explicit formula for $E(x)$.

Of special interest is the class of networks (V,E) where the resistance metric R (on the given vertex vertex-set V) is bounded; see (ii) in Theorem 3 below. This class of networks, for which the diameter of V measured in the resistance metric R is bounded, includes networks having lots of edges with resistors occurring in parallel (Jorgensen and Pearse, 2011).

Theorem 3 *Let $G=(V,E)$, $c:E\to\mathbb{R}_+$ be as above, and let $R^{(c)}:V\times V\to\mathbb{R}_+$ be the resistance-metric in (112). Let M be the metric completion of $\left(V,R^{(c)}\right)$. Then:*

(i) For every $f\in\mathscr{H}$, the function

$$V\ni x\longmapsto f(x)\in\mathbb{C} \tag{122}$$

extends by closure to a uniformly continuous function $\widetilde{f}:M\mapsto\mathbb{C}$.

(ii) If $R^{(c)}$ is assumed bounded, then the RKHS $\mathscr{H}$ is an algebra under point-wise product:

$$(f_1f_2)(x)=f_1(x)\,f_2(x)\,,\quad f_i\in\mathscr{H},\ i=1,2,\ x\in V. \tag{123}$$

(iii) If M is compact, then $\{\widetilde{f}\mid f\in\mathscr{H}\}$ is dense in $C(M)$ in the uniform norm.

Proof The assertions in (i) follow from the following two estimates:

Let $f \in \mathscr{H}$, then

$$|f(x) - f(y)|^2 \leq \|f\|^2_{\mathscr{H}} R^{(c)}(x,y), \quad \forall x, y \in V; \tag{124}$$

and

$$|f(x)| \leq |f(o)| + R^{(c)}(o,x)^{\frac{1}{2}}. \tag{125}$$

The estimates in (124)-(125), in turn, follow from Corollaries 47 and 48.

To prove (ii), we compute the energy-norm of the product $f_1 \cdot f_2$ where $f_i \in \mathscr{H}$, $i = 1, 2$; and we use Corollary 47:

$$\begin{aligned}
& \sum_x \sum_y c_{xy} |f_1(x) f_2(x) - f_1(y) f_2(y)|^2 \\
= & \sum_x \sum_y c_{xy} |(f_1(x) - f_1(y)) f_2(x) + f_1(y)(f_2(x) - f_2(y))|^2 \\
\leq & \sum_x \sum_y c_{xy} \left(|f_1(x) - f_1(y)|^2 + |f_2(x) - f_2(y)|^2\right) \cdot \left(|f_2(x)|^2 + |f_1(y)|^2\right) \\
& \text{(by Schwarz inside)} \\
\leq & \left(\|f_1\|^2_\infty + \|f_2\|^2_\infty\right) \cdot \left(\|f_1\|^2_{\mathscr{H}} + \|f_2\|^2_{\mathscr{H}}\right);
\end{aligned}$$

and we note that the right-side is finite subject to the assumption in (ii).

Proof of (iii): We are assuming here that M is *compact*, and we shall apply the Stone-Weierstrass theorem to the subalgebra

$$\left\{\widetilde{f} \,\middle|\, f \in \mathscr{H}\right\} \subset C(M). \tag{126}$$

Indeed, the conditions for Stone-Weierstrass are satisfied: The functions on LHS in (126) form an algebra, by (ii), closed under complex conjugation; and it separates points in M by Corollary 48. ∎

Example 7 (The binary tree) *Let $A = \{0, 1\}$, and $M := \prod_{\mathbb{N}} A$ the infinite Cartesian product, as a Cantor space. Set $V :=$ all finite words:*

$$V = \bigcup_{n \in \mathbb{N}} \left\{(\alpha_1, \alpha_2, \cdots, \alpha_n) \mid \alpha_i \in \{0, 1\}\right\}; \tag{127}$$

and set $l((\alpha_1, \alpha_2, \cdots, \alpha_n)) =: n$.

For $\omega = (\omega_k)_1^\infty \in M$, set

$$\omega\big|_n := (\omega_1, \omega_2, \cdots, \omega_n) \in V. \tag{128}$$

For two points $\omega, \omega' \in M$, we shall need the number

$$l(\omega \cap \omega') = \sup\left\{n : \omega\big|_n = \omega'\big|_n\right\}. \tag{129}$$

Let $r : \mathbb{N} \to \mathbb{R}_+$ be given such that

$$r(\emptyset) = 0, \quad \sum_{n\in\mathbb{N}} r(n) < \infty. \tag{130}$$

For conductance function $c : E \to \mathbb{R}_+$, set

$$c_{\alpha,(\alpha t)} = \frac{1}{r(l(\alpha))}, \quad \forall \alpha \in V,\ t \in \{0,1\}. \tag{131}$$

One checks that, when (130) holds, then

$$\lim_{n,m\to\infty} R^{(c)}\left(\omega\big|_n, \omega\big|_m\right) = 0.$$

Consider the graph $G_2 = (V, E)$ where the edges are "lines" between α and (αt), where $t \in \{0, 1\}$. See Figure 7.

Lemma 54 *With the settings above, the metric completion $\widetilde{R^{(c)}}$ w.r.t. the resistance metric on V is as follows: For $\omega, \omega' \in M$ (see Figure 9),*

$$\widetilde{R^{(c)}}\left(\omega, \omega'\right) = 2 \sum_{n=l(\omega\cap\omega')}^{\infty} r(n). \tag{132}$$

Let $\mathscr{H}$ be the corresponding energy-Hilbert space $\simeq$ the RKHS of k_c. For $\alpha \in V$, let δ_α be the Dirac-mass at the vertex point α. Then

$$\|\delta_\alpha\|^2_{\mathscr{H}} = \frac{2}{r(l(\alpha))} + \frac{1}{r(l(\alpha)-1)}. \tag{133}$$

(See Figure 8.)

Proof To see this, note that α has the three neighbors sketched in Figure 7, i.e., α^*, $(\alpha 0)$, and $(\alpha 1)$, where α^* is the one-truncated word,

$$\widetilde{R^{(c)}}\left(\omega, \omega'\right) = 2 \sum_{n=l(\omega\cap\omega')}^{\infty} r(n). \tag{134}$$

One checks that when (130) is assumed, then the conditions in point (iii) of the theorem are satisfied. ∎

Corollary 55 *Now return to the discrete restriction of Brownian motion in Section 3.1. Set $V = \{x_1, x_2, x_3, \cdots\}$ where the points $\{x_i\}_{i=1}^{\infty}$ are prescribed such that $x_1 < x_2 < \cdots < x_i < x_{i+1} < \cdots$. We turn V into a weighted graph G as follows: The edges E in G are nearest neighbors; and we define a conductance function $c : E \to \mathbb{R}_+$ by setting*

$$c_{x_i x_{i+1}} := \frac{1}{x_{i+1} - x_i}, \tag{135}$$

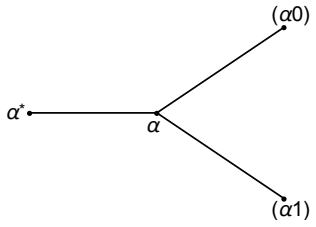


Figure 7: Edges in G_2.

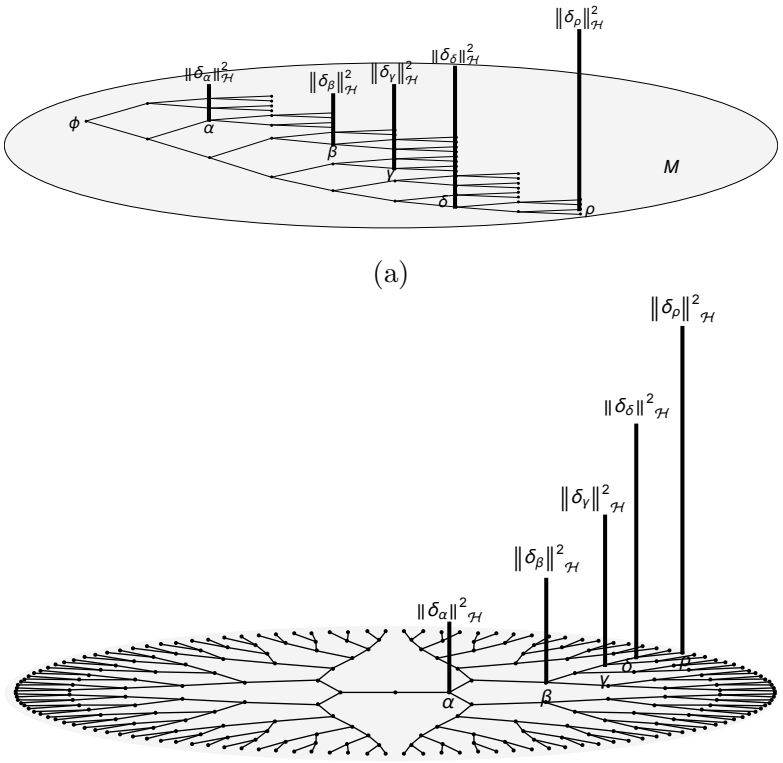


Figure 8: Histogram for $\|\delta_\alpha\|^2_{\mathscr{H}}$ as vertices $\alpha \in V$ approach the boundary. See (133), and note $\|\delta_\alpha\|^2_{\mathscr{H}} \to \infty$ as $\alpha \to M$.

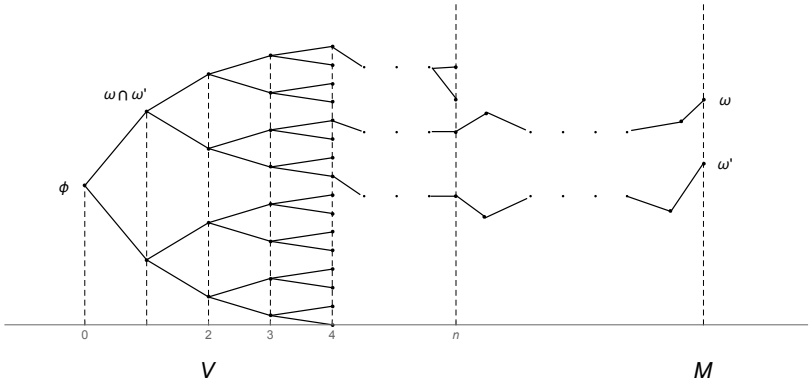


Figure 9: The binary tree and its boundary, the Cantor-set.

and Laplace operator,

$$(\Delta f)(x_i) = \frac{1}{x_{i+1} - x_i}\left(f(x_i) - f(x_{i+1})\right) + \frac{1}{x_i - x_{i-1}}\left(f(x_i) - f(x_{i-1})\right). \tag{136}$$

Then the RKHS associated with the Green's function of Δ in (136) agrees with that from the kernel construction in Section 3.1, i.e., the discrete Cameron-Martin Hilbert space.

Proof Immediate from the previous Proposition and its corollaries. ■

Acknowledgments

The co-authors thank the following colleagues for helpful and enlightening discussions: Professors Daniel Alpay, Sergii Bezuglyi, Ilwoo Cho, Ka Sing Lau, Paul Muhly, Myung-Sin Song, Wayne Polyzou, Gestur Olafsson, Keri Kornelson, and members in the Math Physics seminar at the University of Iowa. We are grateful to the referees for their care to details and for their kind and very helpful suggestions. We have revised following them all.

References

Daniel Alpay and Harry Dym. On reproducing kernel spaces, the Schur algorithm, and interpolation in a general class of domains. In *Operator Theory and Complex Analysis (SApporo, 1991)*, volume 59 of *Oper. Theory Adv. Appl.*, pages 30–77. Birkhäuser, Basel, 1992.

Daniel Alpay and Harry Dym. On a new class of structured reproducing kernel spaces. *J. Funct. Anal.*, 111(1):1–28, 1993.

Daniel Alpay and Palle Jorgensen. Reproducing kernel Hilbert spaces generated by the binomial coefficients. To appear. *ArXiv e-prints*, 2015.

Daniel Alpay, Vladimir Bolotnikov, Aad Dijksma, and Henk de Snoo. On some operator colligations and associated reproducing kernel Hilbert spaces. In *Operator Extensions, Interpolation of Functions and Related Topics*, volume 61 of *Oper. Theory Adv. Appl.*, pages 1–27. Birkhäuser, Basel, 1993.

Daniel Alpay, Palle Jorgensen, Ron Seager, and Dan Volok. On discrete analytic functions: products, rational functions and reproducing kernels. *J. Appl. Math. Comput.*, 41(1-2): 393–426, 2013.

Daniel Alpay, Palle Jorgensen, and Dan Volok. Relative reproducing kernel Hilbert spaces. *Proc. Amer. Math. Soc.*, 142(11):3889–3895, 2014.

Nachman Aronszajn. La théorie des noyaux reproduisants et ses applications. I. *Proc. Cambridge Philos. Soc.*, 39:133–153, 1943.

Nachman Aronszajn. *Reproducing and pseudo-reproducing kernels and their application to the partial differential equations of physics.* Studies in partial differential equations. Technical report 5, preliminary note. Harvard University, Graduate School of Engineering., 1948.

Brighid Boyle, Kristin Cekala, David Ferrone, Neil Rifkin, and Alexander Teplyaev. Electrical resistance of N-gasket fractal networks. *Pacific J. Math.*, 233(1):15–40, 2007.

Ola Bratteli and Palle Jorgensen. *Wavelets Through a Looking Glass.* Applied and Numerical Harmonic Analysis. Birkhäuser Boston, Inc., Boston, MA, 2002.

Andrea Caponnetto, Charles A. Micchelli, Massimiliano Pontil, and Yiming Ying. Universal multi-task kernels. *J. Mach. Learn. Res.*, 9:1615–1646, 2008.

Ilwoo Cho and Palle Jorgensen. Free probability induced by electric resistance networks on energy Hilbert spaces. *Opuscula Math.*, 31(4):549–598, 2011.

Felipe Cucker and Steve Smale. On the mathematical foundations of learning. *Bull. Amer. Math. Soc. (N.S.)*, 39(1):1–49, 2002.

Minh Ha Quang, Sung Ha Kang, and Triet M. Le. Image and video colorization using vector-valued reproducing kernel Hilbert spaces. *J. Math. Imaging Vision*, 37(1):49–65, 2010.

S. Haeseler, M. Keller, D. Lenz, J. Masamune, and M. Schmidt. Global properties of Dirichlet forms in terms of Green's formula. *ArXiv e-prints*, 2014.

Haakan Hedenmalm and Pekka J. Nieminen. The Gaussian free field and Hadamard's variational formula. *Probab. Theory Related Fields*, 159(1-2):61–73, 2014.

Palle Jorgensen and Erin P.J. Pearse. A Hilbert space approach to effective resistance metric. *Complex Anal. Oper. Theory*, 4(4):975–1013, 2010.

Palle Jorgensen and Erin P.J. Pearse. Resistance boundaries of infinite networks. In *Random walks, boundaries and spectra*, volume 64 of *Progr. Probab.*, pages 111–142. Birkhäuser/Springer Basel AG, Basel, 2011.

Palle Jorgensen and Erin P.J. Pearse. A discrete Gauss-Green identity for unbounded Laplace operators, and the transience of random walks. *Israel J. Math.*, 196(1):113–160, 2013.

Samuel Karlin and Zvi Ziegler. Some inequalities of total positivity in pure and applied mathematics. In *Total positivity and its applications (Jaca, 1994)*, volume 359 of *Math. Appl.*, pages 247–261. Kluwer Acad. Publ., Dordrecht, 1996.

Sanjeev Kulkarni and Gilbert Harman. *An elementary introduction to statistical learning theory.* Wiley Series in Probability and Statistics. John Wiley & Sons, Inc., Hoboken, NJ, 2011.

Sneh Lata and Vern Paulsen. The Feichtinger conjecture and reproducing kernel Hilbert spaces. *Indiana Univ. Math. J.*, 60(4):1303–1317, 2011.

Yi Lin and Lawrence D. Brown. Statistical properties of the method of regularization with periodic Gaussian reproducing kernel. *Ann. Statist.*, 32(4):1723–1743, 2004.

Edward Nelson. Kernel functions and eigenfunction expansions. *Duke Math. J.*, 25:15–27, 1957.

Kasso A. Okoudjou and Robert S. Strichartz. Weak uncertainty principles on fractals. *J. Fourier Anal. Appl.*, 11(3):315–331, 2005.

Kasso A. Okoudjou, Robert S. Strichartz, and Elizabeth K. Tuley. Orthogonal polynomials on the Sierpinski gasket. *Constr. Approx.*, 37(3):311–340, 2013.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning.* Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA, 2006.

Bernhard Schlkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond (Adaptive Computation and Machine Learning).* The MIT Press, 1st edition, 12 2001.

Oded Schramm and Scott Sheffield. A contour line of the continuum Gaussian free field. *Probab. Theory Related Fields*, 157(1-2):47–80, 2013.

John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis.* Cambridge University Press, 2004.

Steve Smale and Ding-Xuan Zhou. Online learning with Markov sampling. *Anal. Appl. (Singap.)*, 7(1):87–113, 2009.

Ingo Steinwart. On the influence of the kernel on the consistency of support vector machines. *J. Mach. Learn. Res.*, 2:67–93, March 2002.

Robert S. Strichartz. Transformation of spectra of graph Laplacians. *Rocky Mountain J. Math.*, 40(6):2037–2062, 2010.

Robert S. Strichartz and Alexander Teplyaev. Spectral analysis on infinite Sierpiński fractafolds. *J. Anal. Math.*, 116:255–297, 2012.

Mirjana Vuletić. The Gaussian free field and strict plane partitions. In *25th International Conference on Formal Power Series and Algebraic Combinatorics (FPSAC 2013)*, Discrete Math. Theor. Comput. Sci. Proc., AS, pages 1041–1052. Assoc. Discrete Math. Theor. Comput. Sci., Nancy, 2013.

Haizhang Zhang, Yuesheng Xu, and Qinghui Zhang. Refinement of operator-valued reproducing kernels. *J. Mach. Learn. Res.*, 13:91–136, 2012.

Journal of Machine Learning Research 16 (2015) 3115-3150 Submitted 11/13; Revised 3/15; Published 12/15

A Direct Estimation of High Dimensional Stationary Vector Autoregressions

Fang Han FHAN@JHU.EDU
Department of Biostatistics
Johns Hopkins University
Baltimore, MD 21205, USA

Huanran Lu HUANRANL@PRINCETON.EDU
Han Liu HANLIU@PRINCETON.EDU
Department of Operations Research and Financial Engineering
Princeton University
Princeton, NJ 08544, USA

Editor: Xiaotong Shen

Abstract

The vector autoregressive (VAR) model is a powerful tool in learning complex time series and has been exploited in many fields. The VAR model poses some unique challenges to researchers: On one hand, the dimensionality, introduced by incorporating multiple numbers of time series and adding the order of the vector autoregression, is usually much higher than the time series length; On the other hand, the temporal dependence structure naturally present in the VAR model gives rise to extra difficulties in data analysis. The regular way in cracking the VAR model is via "least squares" and usually involves adding different penalty terms (e.g., ridge or lasso penalty) in handling high dimensionality. In this manuscript, we propose an alternative way in estimating the VAR model. The main idea is, via exploiting the temporal dependence structure, formulating the estimating problem to a linear program. There is instant advantage of the proposed approach over the lasso-type estimators: The estimation equation can be decomposed to multiple sub-equations and accordingly can be solved efficiently using parallel computing. Besides that, we also bring new theoretical insights into the VAR model analysis. So far the theoretical results developed in high dimensions (e.g., Song and Bickel, 2011 and Kock and Callot, 2015) are based on stringent assumptions that are not transparent. Our results, on the other hand, show that the spectral norms of the transition matrices play an important role in estimation accuracy and build estimation and prediction consistency accordingly. Moreover, we provide some experiments on both synthetic and real-world equity data. We show that there are empirical advantages of our method over the lasso-type estimators in parameter estimation and forecasting.

Keywords: transition matrix, multivariate time series, vector autoregressive model, double asymptotic framework, linear program

1. Introduction

The vector autoregressive (VAR) model plays a fundamental role in analyzing multivariate time series data and has many applications in numerous academic fields. The VAR model is heavily used in finance (Tsay, 2005), econometrics (Sims, 1980), and brain imaging data

analysis (Valdés-Sosa et al., 2005). For example, in understanding the brain connectivity network, multiple resting-state functional magnetic resonance imaging (rs-fMRI) data are obtained by consecutively scanning the same subject for approximately a hundred times or more. This naturally produces a high dimensional dependent data and a common strategy in handling such data is via building a vector autoregressive model (see Qiu et al., and the references therein).

This manuscript considers estimating the VAR model. Our focus is on the stationary vector autoregression with the order (or called lag) p and Gaussian noises. More specifically, let random vectors $X_1, \ldots, X_T$ be from a stochastic process $(X_t)_{t=-\infty}^{\infty}$. Each X_t is a d-dimensional random vector and satisfies that

$$X_t = \sum_{k=1}^{p} A_k^{\mathrm{T}} X_{t-k} + Z_t, \quad Z_t \sim N_d(0, \Psi),$$

where $A_1, \ldots, A_p$ are called the transition matrices and $(Z_t)_{t=-\infty}^{\infty}$ are independent multivariate Gaussian noises. Via assuming $\det(I_d - \sum_{k=1}^{p} A_k^{\mathrm{T}} z^k) \neq 0$ for all $z \in \mathcal{C}$ with modulus not greater than one, we then have the process is stationary (check, for example, Section 2.1 in Lütkepohl, 2005) and $X_t \sim N_d(0, \Sigma)$ for some covariance matrix Σ depending on $\{A_k, k = 1, \ldots, p\}$ and Ψ.

There are in general three main targets in analyzing an VAR model. One is to estimate the transition matrices $A_1, \ldots, A_p$. These transition matrices reveal the temporal dependence in the data sequence and estimating them builds a fundamental first step in forecasting. Moreover, the zero and nonzero entries in the transition matrices directly incorporate the Granger non-causalities and causalities with regard to the stochastic sequence (see, for example, Corollary 2.2.1 in Lütkepohl, 2005). Another one of interest is the error covariance Ψ, which reveals the contemporaneous interactions among d time series. Finally, by merely treating the temporal dependence as another measure of the data dependence (in parallel to the mixing conditions, Bradley, 2005), it is also of interest to estimate the covariance matrix Σ.

This manuscript focuses on estimating the transition matrices $A_1, \ldots, A_p$, while noting that the techniques developed here can also be exploited to estimate the covariance matrix Σ and the noise covariance Ψ. We first review the methods developed so far in transition matrix estimation. Let $A = (A_1^{\mathrm{T}}, \ldots, A_p^{\mathrm{T}})^{\mathrm{T}} \in \mathbb{R}^{dp \times d}$ be the combination of the transition matrices. Given $X_1, \ldots, X_T$, the perhaps most classic method in estimating A is least squares minimization (Hamilton, 1994)

$$\widehat{A}^{\mathrm{LSE}} = \underset{M \in \mathbb{R}^{dp \times d}}{\operatorname{argmin}} \|\widetilde{Y} - M^{\mathrm{T}} \widetilde{X}\|_{\mathsf{F}}^2, \tag{1}$$

where $\|\cdot\|_{\mathsf{F}}$ is the matrix Frobenius norm, $\widetilde{Y} = (X_{p+1}, \ldots, X_T) \in \mathbb{R}^{d \times (T-p)}$, and $\widetilde{X} = \{(X_p^{\mathrm{T}}, \ldots, X_1^{\mathrm{T}})^{\mathrm{T}}, \ldots, (X_{T-1}^{\mathrm{T}}, \ldots, X_{T-p}^{\mathrm{T}})^{\mathrm{T}}\} \in \mathbb{R}^{(dp) \times (T-p)}$. However, a fatal problem in (1) is that the product of the order of the autoregression p and the number of time series d is frequently larger than the time series length T. Therefore, the model has to be constrained to enforce identifiability. A common strategy is to add sparsity on the transition matrices so that the number of nonzero entries is less than T. Built on this assumption, there has been a large literature discussing adding different penalty terms to (1) for regularizing

the estimator: From the ridge-penalty to the lasso-penalty and more non-concave penalty terms. In the following we list the major efforts. Hamilton (1994) discussed the use of the ridge-penalty $\|M\|_{\mathsf{F}}^2$ in estimating the transition matrices. Hsu et al. (2008) proposed to add the L_1-penalty in estimating the transition matrices, inducing a sparse output. Several extensions to transition matrix estimation in the VAR model include: Wang et al. (2007) exploited the L_1-penalty in simultaneously estimating the regression coefficients and determining the number of lags in a linear regression model with autoregressive errors. In detecting causality, Haufe et al. (2008) transferred the problem to estimating transition matrices in an VAR model and advocated using a group-lasso penalty for inducing joint sparsity among a whole block of coefficients. In studying the graphical Granger causality problem, Shojaie and Michailidis (2010) exploited the VAR model and proposed to estimate the coefficients using a truncated weighted L_1-penalty. Song and Bickel (2011) exploited the L_1 penalty in a complicated VAR model and aimed to select the variables and lags simultaneously.

The theoretical properties of the L_1-regularized estimator have been analyzed in Bento et al. (2010), Nardi and Rinaldo (2011), Song and Bickel (2011), and Kock and Callot (2015) under the assumption that the matrix A is sparse, i.e., the number of nonzero entries in A is much less than the dimension of parameters pd^2. Nardi and Rinaldo (2011) provided both subset and parameter estimation consistency results under a relatively low dimensional settings with $d = o(n^{1/2})$. Bento et al. (2010) studied the problem of estimating supports sets of the transition matrices in the high dimensional settings and proposed an "irrepresentable condition" similar as what is proposed in the linear regression model (Zou, 2006; Zhao and Yu, 2006; Meinshausen and Bühlmann, 2006; Wainwright, 2009). It is for the L_1 regularized estimator to attain the support set selection consistency. In parallel, Song and Bickel (2011) and Kock and Callot (2015) studied the parameter estimation and support set selection consistency of the L_1-regularized estimator in high dimensions.

In this paper, we propose a new approach to estimate the transition matrix A. Different from the line of lasso-based estimation procedures, which are built on penalizing the least square term, we exploit the linear programming technique and the proposed method is very fast to solve via parallel computing. Moreover, we do not need A to be exactly sparse and allow it to be only "weakly sparse". The main idea is to estimate A using the relationship between A and the marginal and lag 1 autocovariance matrices (such a relationship is referred to as the Yule-Walker equation). We thus formulate the estimation procedure to a linear program, while adding the $\|\cdot\|_{\max}$ (element-wise supremum norm) for model identifiability. Here we note that the proposed procedure can be considered as a generalization of the Dantzig selector (Candes and Tao, 2007) to the linear regression model with multivariate response. Indeed, our proposed method can also be exploited in conducting multivariate regression (Breiman and Friedman, 1997).

The proposed method enjoys several advantages compared to the existing ones: (i) Computationally, our method can be formulated into d linear programs and can be solved in parallel. Similar ideas have been used in learning high dimensional linear regression (Candes and Tao, 2007; Bickel et al., 2009) and graphical models (Yuan, 2010; Cai et al., 2011). (ii) In the model-level, our method allows A to be only weakly sparse. (iii) Theoretically, so far the analysis on lasso-type estimators (Song and Bickel, 2011; Kock and Callot, 2015) depends on certain regularity conditions, restricted eigenvalue conditions on the design

matrix for example, which are not transparent and do not explicitly reveal the role of temporal dependence in it. In contrast, we provide explicit nonasymptotic analysis, and our analysis highlights the spectral norm $\|A\|_2$ in estimation accuracy, which is inspired by some recent developments (Loh and Wainwright, 2012). Moreover, for exact sign recovery, our analysis does not need the "irrepresentable condition" which is usually required in the analysis of lasso-type estimators (Bento et al., 2010).

The major theoretical results are briefly stated as follows. We adopt a double asymptotic framework where d is allowed to increase with T. We call a matrix s-sparse if there are at most s nonzero elements on each of its column. Under mild conditions, we provide the explicit rates of convergence of our estimator $\widehat{A}$ based on the assumption that A is s-sparse (Cai et al., 2011). In particular, for lag 1 time series, we show that

$$\|\widehat{A} - A\|_1 = O_P\left\{\frac{s\|A\|_1}{1-\|A\|_2}\left(\frac{\log d}{T}\right)^{1/2}\right\}, \quad \|\widehat{A} - A\|_{\max} = O_P\left\{\frac{\|A\|_1}{1-\|A\|_2}\left(\frac{\log d}{T}\right)^{1/2}\right\},$$

where $\|\cdot\|_{\max}$ and $\|\cdot\|_q$ represent the matrix elementwise absolute maximum norm ($L_{\max}$ norm) and induced L_q norm (detailed definitions will be provided in §2). Using the $L_{\max}$ norm consistency result, we further provide the sign recovery consistency of the proposed method. This result is of self interest and sheds light to detecting Granger causality. We also provide the prediction consistency results based on the L_1 consistency result and show that element-wise error in prediction can be controlled. Here for simplicity we only provide the results when A is exactly sparse and defer the presentation of the results for weakly sparse matrix to Section 4.

The rest of the paper is organized as follows. In Section 2, we briefly review the vector autoregressive model. In Section 3, we introduce the proposed method for estimating the transition matrices of the vector autoregressive model. In Section 4, we provide the main theoretical results. In Section 5, we apply the new method to both synthetic and real equity data for illustrating its effectiveness. More discussions are provided in the last section. Detailed technical proofs are provided in the appendix[1].

2. Background

In this section, we briefly review the vector autoregressive model. Let $M = (M_{jk}) \in \mathbb{R}^{d\times d}$ and $v = (v_1, ..., v_d)^T \in \mathbb{R}^d$ be a matrix and an vector of interest. We denote v_I to be the subvector of v whose entries are indexed by a set $I \subset \{1, \ldots, d\}$. We also denote $M_{I,J}$ to be the submatrix of M whose rows are indexed by I and columns are indexed by J. We denote $M_{I,*}$ to be the submatrix of M whose rows are indexed by I, $M_{*,J}$ to be the submatrix of M whose columns are indexed by J. For $0 < q < \infty$, we define the L_0, L_q, and L_∞ vector (pseudo-)norms to be

$$\|v\|_0 := \sum_{j=1}^{d} I(v_j \neq 0), \quad \|v\|_q := \Big(\sum_{j=1}^{d} |v_j|^q\Big)^{1/q}, \quad \text{and} \quad \|v\|_\infty := \max_{1\leq j\leq d} |v_j|,$$

1. Some of the results in this paper were first stated without proof in a conference version (Han and Liu, 2013).

where $I(\cdot)$ is the indicator function. Letting M be a matrix, we denote the matrix L_q, $L_{\max}$, and Frobenius norms to be

$$\|M\|_q := \max_{\|v\|_q=1} \|Mv\|_q, \quad \|M\|_{\max} := \max_{jk} |M_{jk}|, \text{ and } \|M\|_{\mathsf{F}} := \Big(\sum_{j,k} |M_{jk}|^2\Big)^{1/2}.$$

We denote $\mathbf{1}_d = (1, \ldots, 1)^{\mathrm{T}} \in \mathbb{R}^d$. Let $\sigma_1(M) \geq \cdots \geq \sigma_d(M)$ be the singular values of M.

Let $p \geq 1$ be an integer. A lag p vector autoregressive process can be elaborated as follows: Let $(X_t)_{t=-\infty}^{\infty}$ be a stationary sequence of random vectors in $\mathbb{R}^d$ with mean 0 and covariance matrix Σ. We say that $(X_t)_{t=-\infty}^{\infty}$ follow a lag p vector autoregressive model if and only if they satisfy

$$X_t = \sum_{k=1}^{p} A_k^{\mathrm{T}} X_{t-k} + Z_t \quad (t \in \mathbb{Z}). \tag{2}$$

Here $A_1, \ldots, A_p$ are called transition matrices. We denote $A = (A_1^{\mathrm{T}}, \ldots, A_p^{\mathrm{T}})^{\mathrm{T}}$ to be the combination of the transition matrices. We assume that Z_t are independently and identically generated from a Gaussian distribution $N_d(0, \Psi)$. Moreover, Z_t and $(X_s)_{s<t}$ are independent for any $t \in \mathbb{Z}$. We pose an additional assumption that $\det(I_d - \sum_{k=1}^{p} A_k^{\mathrm{T}} z^k) \neq 0$ for all $z \in \mathcal{C}$ with modulus not greater than one. This guarantees that the sequence is stationary and we have, for any $t \in \mathbb{Z}$, X_t follows a Gaussian distribution $N_d(0, \Sigma)$,

We denote $\Sigma_i(\cdot)$ to be an operator on the process $(X_t)_{t=-\infty}^{\infty}$. In particular, we define $\Sigma_i\{(X_t)\} = \mathrm{Cov}(X_0, X_i)$. It is easy to see that $\Sigma_0\{(X_t)\} = \Sigma$. If the lag of the vector autoregressive model is 1 (i.e., $X_t = A_1^{\mathrm{T}} X_{t-1} + Z_t$, for any $t \in \mathbb{Z}$), by simple calculation we have the so called "Yule-Walker Equation"

$$\Sigma_i\{(X_t)\} = \Sigma_0\{(X_t)\}(A_1)^i, \tag{3}$$

which further implies that

$$A_1 = [\Sigma_0\{(X_t)\}]^{-1} \cdot \Sigma_1\{(X_t)\}.$$

The results for lag 1 vector autoregressive model can be extended to the lag p vector autoregressive model by appropriately redefining the random vectors. In detail, the autoregressive model with lag p shown in (2) can be reformulated as an autoregressive model with lag 1

$$\widetilde{X}_t = \widetilde{A}^{\mathrm{T}} \widetilde{X}_{t-1} + \widetilde{Z}_t, \tag{4}$$

where

$$\widetilde{X}_t = \begin{pmatrix} X_{t+p-1} \\ X_{t+p-2} \\ \vdots \\ X_t \end{pmatrix}, \quad \widetilde{A} = \begin{pmatrix} A_1 & I_d & 0 & \ldots & 0 \\ \vdots & \ddots & \ldots & \ldots & \vdots \\ A_{p-1} & 0 & 0 & \ldots & I_d \\ A_p & 0 & 0 & \ldots & 0 \end{pmatrix}, \quad \widetilde{Z}_t = \begin{pmatrix} Z_{t+p-1} \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{5}$$

Here $I_d \in \mathbb{R}^{d \times d}$ is the identity matrix, $\widetilde{X}_t \sim N_{dp}(0, \widetilde{\Sigma})$ for $t = 1, \ldots, T$, and $\widetilde{Z}_t \sim N_{dp}(0, \widetilde{\Psi})$ with $\widetilde{\Sigma} = \mathrm{Cov}(\widetilde{X}_t)$ and $\widetilde{\Psi} = \mathrm{Cov}(\widetilde{Z}_t)$. Therefore, we also have

$$\widetilde{A} = [\Sigma_0\{(\widetilde{X}_t)\}]^{-1} \cdot \Sigma_1\{(\widetilde{X}_t)\}. \tag{6}$$

This is similar to the relationship for the lag 1 vector autoregressive model.

3. Methods and Algorithms

We provide a new formulation to estimate $A_1, \ldots, A_p$ for the vector autoregressive model. Let $X_1, \ldots, X_T$ be from a lag p vector autoregressive process $(X_t)_{t=-\infty}^{\infty}$ and we denote $\widetilde{X}_t = (X_{t+p-1}^{\mathrm{T}}, \ldots, X_t^{\mathrm{T}})^{\mathrm{T}}$ for $t = 1, \ldots, T-p+1$. We denote S and S_1 to be the marginal and lag 1 sample covariance matrices of $(\widetilde{X}_t)_{t=1}^{T-p+1}$

$$S := \frac{1}{T-p+1} \sum_{t=1}^{\mathrm{T}-p+1} \widetilde{X}_t \widetilde{X}_t^{\mathrm{T}}, \quad S_1 := \frac{1}{T-p} \sum_{t=1}^{T-p} \widetilde{X}_t \widetilde{X}_{t+1}^{\mathrm{T}}. \tag{7}$$

Using the connection between $\widetilde{A}$ and $\Sigma_0\{(\widetilde{X}_t)\}, \Sigma_1\{(\widetilde{X}_t)\}$ shown in (6), we know that a good estimator $\check{\Omega}$ of $\widetilde{A}$ shall satisfy that

$$\|\Sigma_0\{(\widetilde{X}_t)\}\check{\Omega} - \Sigma_1\{(\widetilde{X}_t)\}\| \tag{8}$$

is small enough with regard to a certain matrix norm $\|\cdot\|$. Moreover, using the fact that $A = (A_1^{\mathrm{T}}, \ldots, A_p^{\mathrm{T}})^{\mathrm{T}} = \widetilde{A}_{*,J}$, where $J = \{1, \ldots, d\}$, by (8) we have that a good estimate $\check{A}$ of A shall satisfy

$$\|\Sigma_0\{(\widetilde{X}_t)\}\check{A} - [\Sigma_1\{(\widetilde{X}_t)\}]_{*,J}\| \tag{9}$$

is small enough.

Motivated by (9), we estimate $A_1, \ldots, A_p$ via replacing $\Sigma_0\{(\widetilde{X}_t)$ and $[\Sigma_1\{(\widetilde{X}_t)\}]_{*,J}$ with their empirical versions. For formulating the estimation equation to a linear program, we use the $L_{\max}$ norm. Accordingly, we end in solving the following convex optimization program

$$\widehat{\Omega} = \underset{M \in \mathbb{R}^{dp \times d}}{\operatorname{argmin}} \sum_{jk} |M_{jk}|, \quad \text{subject to} \quad \|SM - (S_1)_{*,J}\|_{\max} \le \lambda_0, \tag{10}$$

where $\lambda_0 > 0$ is a tuning parameter. In (10), the constraint part aims to find an estimate that approximates the true parameter well, and combined with the minimization part, aims to induce certain sparsity. Let $\widehat{\Omega}_{*,j} = \widehat{\beta}_j$, it is easy to see that (10) can be decomposed to many subproblems and each $\widehat{\beta}_j$ can be solved by

$$\widehat{\beta}_j = \underset{v \in \mathbb{R}^{dp}}{\operatorname{argmin}} \|v\|_1, \quad \text{subject to} \quad \|Sv - (S_1)_{*,j}\|_\infty \le \lambda_0. \tag{11}$$

Accordingly, compared to the lasso-type procedures, the proposed method can be solved in parallel and therefore is computationally more efficient.

Once $\widehat{\Omega}$ is obtained, the estimator of the transition matrix A_k can then be written as

$$\widehat{A}_k = \widehat{\Omega}_{J_k,*}, \tag{12}$$

where we denote $J_k = \{j : d(k-1)+1 \le j \le dk\}$.

We now show that the optimization in (11) can be formulated into a linear program. Recall that any real number a takes the decomposition $a = a^+ - a^-$, where $a^+ = a \cdot I(a \ge 0)$ and $a^- = -a \cdot I(a < 0)$. For any vector $v = (v_1, \ldots, v_d)^{\mathrm{T}} \in \mathbb{R}^d$, let $v^+ = (v_1^+, \ldots, v_d^+)^{\mathrm{T}}$ and $v^- = (v_1^-, \ldots, v_d^-)^{\mathrm{T}}$. We denote $v \ge 0$ if $v_1, \ldots, v_d \ge 0$ and $v < 0$ if $v_1, \ldots, v_d < 0$, $v_1 \ge v_2$

if $v_1 - v_2 \geq 0$, and $v_1 < v_2$ if $v_1 - v_2 < 0$. Letting $v = (v_1, \ldots, v_d)^{\mathrm{T}}$, the problem in (11) can be further relaxed to the following problem

$$\begin{aligned} \widehat{\beta}_j &= \underset{v^+, v^-}{\operatorname{argmin}} \mathbf{1}_d^{\mathrm{T}}(v^+ + v^-), \\ \text{subject to } \|Sv^+ - Sv^- - (S_1)_{*,j}\|_\infty &\leq \lambda_0, \quad v^+ \geq 0, v^- \geq 0. \end{aligned} \tag{13}$$

To minimize $\mathbf{1}_d^{\mathrm{T}}(v^+ + v^-)$, v^+ or v^- can not be both nonzero. Therefore, the solution to (13) is exactly the solution to (11). The optimization in (13) can be written as

$$\begin{aligned} \widehat{\beta}_j &= \underset{v^+, v^-}{\operatorname{argmin}} \mathbf{1}_d^{\mathrm{T}}(v^+ + v^-), \\ \text{subject to} \quad Sv^+ - Sv^- - (S_1)_{*,j} &\leq \lambda_0 \mathbf{1}_d, \\ -Sv^+ + Sv^- + (S_1)_{*,j} &\leq \lambda_0 \mathbf{1}_d, \\ v^+ \geq 0, v^- &\geq 0. \end{aligned}$$

This is equivalent to

$$\widehat{\beta}_j = \underset{\omega}{\operatorname{argmin}} \mathbf{1}_{2d}^{\mathrm{T}} \omega, \quad \text{subject to} \quad \theta + W\omega \geq 0, \quad \omega \geq 0, \tag{14}$$

where

$$\omega = \begin{pmatrix} v^+ \\ v^- \end{pmatrix}, \quad \theta = \begin{bmatrix} (S_1)_{*,j} + \lambda_0 \mathbf{1}_d \\ -(S_1)_{*,j} + \lambda_0 \mathbf{1}_d \end{bmatrix}, \quad W = \begin{pmatrix} -S & S \\ S & -S \end{pmatrix}.$$

The optimization (14) is a linear program. We can solve it using the simplex algorithm (Murty, 1983).

4. Theoretical Properties

In this section, under the double asymptotic framework, we provide the nonasymptotic rates of convergence in parameter estimation under the matrix L_1 and $L_{\max}$ norms.

We first present the rates of convergence of the estimator $\widehat{\Omega}$ in (10) under the vector autoregressive model with lag 1. This result allows us to sharply characterize the impact of the temporal dependence of the time series on the obtained rate of convergence. In particular, we show that the rate of convergence is closely related to the L_1 and L_2 norms of the transition matrix A_1, where $\|A_1\|_2$ is the key part in characterizing the impact of temporal dependence on estimation accuracy. Secondly, we present the sign recovery consistency result of our estimator. Compared to the lasso-type estimators, our result does not require the irrepresentable condition. These results are combined together to show that we have the prediction consistency, i.e., the term $\|A_1 X_T - \widehat{A}_1 X_T\|$ goes to zero with regard to certain norms $\|\cdot\|$. The application to lag $p > 1$ case is left for future studies.

We start with some additional notation. Let $M_d \in \mathbb{R}$ be a quantity which may scale with the time series length and dimension (T, d). We define the set of square matrices in $\mathbb{R}^{d \times d}$, denoted by $\mathcal{M}(q, s, M_d)$, as

$$\mathcal{M}(q, s, M_d) := \Big\{ M \in \mathbb{R}^{d \times d} : \max_{1 \leq j \leq d} \sum_{i=1}^{d} |M_{ij}|^q \leq s, \|M\|_1 \leq M_d \Big\}.$$

For $q = 0$, the class $\mathcal{M}(0, s, M_d)$ contains all the s-sparse matrices with bounded L_1 norms.

There are two general remarks about the model $\mathcal{M}(q, s, M_d)$: (i) $\mathcal{M}(q, s, M_d)$ can be considered as the matrix version of the vector "weakly sparse set" explored in Raskutti et al. (2011) and Vu and Lei (2012). Such a way to define the weakly sparse set of matrices is also investigated in Cai et al. (2011). (ii) For the exactly sparse matrix set, $\mathcal{M}(0, s, M_d)$, the sparsity level s here represents the largest number of nonzero entries in each column of the matrix. In contrast, the sparsity level s' exploited in Kock and Callot (2015) is the total number of nonzero entries in the matrix. We must have $s' \geq s$ and regularly $s' \gg s$ (means $s/s' \to 0$).

The next theorem presents the L_1 and $L_{\max}$ rates of convergence of our estimator under the vector autoregressive model with lag 1.

Theorem 1 *Suppose that $(X_t)_{t=1}^T$ are from a lag 1 vector autoregressive process $(X_t)_{t=-\infty}^{\infty}$ as described in (2). We assume the transition matrix $A_1 \in \mathcal{M}(q, s, M_d)$ for some $0 \leq q < 1$. Let $\widehat{A}_1$ be the optimum to (10) with the tuning parameter*

$$\lambda_0 = \frac{32\|\Sigma\|_2 \max_j(\Sigma_{jj})}{\min_j(\Sigma_{jj})(1 - \|A\|_2)}(2M_d + 3)\left(\frac{\log d}{T}\right)^{1/2}.$$

For $T \geq 6 \log d + 1$ and $d \geq 8$, we have, with probability no smaller than $1 - 14d^{-1}$,

$$\|\widehat{A}_1 - A_1\|_1 \leq 4s\left\{\frac{32\|\Sigma^{-1}\|_1 \max_j(\Sigma_{jj})\|\Sigma\|_2}{\min_j(\Sigma_{jj})(1 - \|A_1\|_2)}(2M_d + 3)\left(\frac{\log d}{T}\right)^{1/2}\right\}^{1-q}. \tag{15}$$

Moreover, with probability no smaller than $1 - 14d^{-1}$,

$$\|\widehat{A}_1 - A_1\|_{\max} \leq \frac{64\|\Sigma^{-1}\|_1 \max_j(\Sigma_{jj})\|\Sigma\|_2}{\min_j(\Sigma_{jj})(1 - \|A_1\|_2)}(2M_d + 3)\left(\frac{\log d}{T}\right)^{1/2}. \tag{16}$$

In the above results, Σ is the marginal covariance matrix of X_t.

It can be observed that, similar to the lasso and Dantzig selector (Candes and Tao, 2007; Bickel et al., 2009), the tuning parameter λ_0 here depends on the variance term Σ. In practice, same as most preceded developments (see, for example, Song and Bickel, 2011), we can use a data-driven way to select the tuning parameter. In this manuscript we explore using cross-validation to choose λ_0 with the best prediction accuracy. In Section 5 we will show that the procedure of selecting the tuning parameter via cross-validation gives reasonable results.

Here A_1 is assumed to be at least weakly sparse and belong to the set $\mathcal{M}(q, s, M_d)$. This is merely for the purpose of model identifiability. Otherwise, we will have multiple global optima in the optimization problem.

The obtained rates of convergence in Theorem 1 depend on both Σ and A_1 with $\|A_1\|_2$ characterizing the temporal dependence. In particular, the estimation error is related to the spectral norm of the transition matrix A_1. Intuitively, this is because $\|A_1\|_2$ characterizes the data dependence of $X_1, \ldots, X_T$, and accordingly intrinsically characterizes how much information there is in the data. If $\|A_1\|_2$ is larger, then there is less information we can

exploit in estimating A_1. Technically, $\|A_1\|_2$ determines the rate of convergence of S and S_1 to their population counterparts. We refer to the proofs of Lemmas 1 and 2 for details.

In the following, we list two examples to provide more insights about the results in Theorem 1.

Example 1 *We consider the case where Σ is a strictly diagonal dominant (SDD) matrix (Horn and Johnson, 1990) with the property*

$$\delta_i := |\Sigma_{ii}| - \sum_{j\neq i} |\Sigma_{ij}| \geq 0, \quad (i = 1, \ldots, d).$$

This corresponds to the cases where the d entries in any X_t with $t \in \{1, \ldots, T\}$ are weakly dependent. In this setting, Ahlberg and Nilson (1963) showed that

$$\|\Sigma^{-1}\|_1 = \|\Sigma^{-1}\|_\infty \leq \Big\{\min_i \Big(|\Sigma_{ii}| - \sum_{j\neq i} |\Sigma_{ij}|\Big)\Big\}^{-1} = \max_i(\delta_i^{-1}). \tag{17}$$

Moreover, by algebra, we have

$$\|\Sigma\|_2 \leq \|\Sigma\|_1 = \max_i \Big(|\Sigma_{ii}| + \sum_{j\neq i} |\Sigma_{ij}|\Big) \leq 2 \max_i(|\Sigma_{ii}|). \tag{18}$$

Equations (17) *and* (18) *suggest that, when $\max_i(\Sigma_{ii})$ is upper bounded, and both $\min_i(\Sigma_{ii})$ and δ_i are lower bounded by a fixed constant, we have both $\|\Sigma^{-1}\|_1$ and $\|\Sigma\|_2$ are upper bounded, and the obtained rates of convergence in* (15) *and* (16) *can be simplified as*

$$\|\widehat{A}_1 - A_1\|_1 = O_P\left[s\left\{\frac{M_d}{1-\|A_1\|_2}\left(\frac{\log d}{T}\right)^{1/2}\right\}^{1-q}\right],$$
$$\|\widehat{A}_1 - A_1\|_{\max} = O_P\left\{\frac{M_d}{1-\|A_1\|_2}\left(\frac{\log d}{T}\right)^{1/2}\right\}.$$

Example 2 *We can generalize the "entry-wise weakly dependent" structure in Example 1 to a "block-wise weakly dependent" structure. More specifically, we consider the case where $\Sigma = (\Sigma_{jk}^b)$ with blocks $\Sigma_{jk}^b \in \mathbb{R}^{d_j \times d_k}$ $(1 \leq j \leq K)$ is a strictly block diagonal dominant (SBDD) matrix with the property*

$$\delta_i^b = \|(\Sigma_{ii}^b)^{-1}\|_\infty^{-1} - \sum_{j\neq i} \|\Sigma_{ij}^b\|_\infty > 0 \quad (i = 1, \ldots, K).$$

In this case, Varah (1975) showed that

$$\|\Sigma^{-1}\|_1 = \|\Sigma^{-1}\|_\infty \leq \Big\{\min_i \Big(\|(\Sigma_{ii}^b)^{-1}\|_\infty^{-1} - \sum_{j\neq i} \|\Sigma_{ij}^b\|_\infty\Big)\Big\}^{-1} = \max\{(\delta_i^b)^{-1}\}.$$

Moreover, we have

$$\|\Sigma\|_2 \leq \|\Sigma\|_1 \leq \max_i(\|(\Sigma_{ii}^b)^{-1}\|_\infty^{-1} + \|\Sigma_{ii}^b\|_\infty).$$

Accordingly, generally $(\|(\Sigma^b_{ii})^{-1}\|^{-1}_\infty + \|\Sigma^b_{ii}\|_\infty)$ *is in the scale of* $\max_i(d_i) \ll d$*, and when* δ^b_i *are lower bounded and the condition number of* Σ *is upper bounded, we have the obtained rates of convergence can be simplified as*

$$\|\widehat{A}_1 - A_1\|_1 = O_P\left[s\left\{\frac{M_d \cdot \max_i(d_i)}{1-\|A_1\|_2}\left(\frac{\log d}{T}\right)^{1/2}\right\}^{1-q}\right],$$

$$\|\widehat{A}_1 - A_1\|_{\max} = O_P\left\{\frac{M_d \cdot \max_i(d_i)}{1-\|A_1\|_2}\left(\frac{\log d}{T}\right)^{1/2}\right\}.$$

We then continue to the results of feature selection. If we have $A_1 \in \mathcal{M}(0, s, M_d)$, from the element-wise $L_{\max}$ norm convergence, a sign recovery result can be obtained. In detail, let $\breve{A}_1$ be a truncated version of $\widehat{A}_1$ with level γ

$$(\breve{A}_1)_{ij} = (\widehat{A}_1)_{ij} I\{|(\widehat{A}_1)_{ij}| \geq \gamma\}. \tag{19}$$

The following corollary shows that $\breve{A}_1$ recovers the sign of A_1 with overwhelming probability.

Corollary 1 *Suppose that the conditions in Theorem 1 hold and* $A_1 \in \mathcal{M}(0, s, M_d)$*. If we choose the truncation level*

$$\gamma = \frac{64\|\Sigma^{-1}\|_1 \max_j(\Sigma_{jj})\|\Sigma\|_2}{\min_j(\Sigma_{jj})(1-\|A_1\|_2)}(2M_d+3)\left(\frac{\log d}{T}\right)^{1/2}$$

in (19) *and with the assumption that*

$$\min_{\{(j,k):(A_1)_{jk}\neq 0\}} |(A_1)_{jk}| \geq 2\gamma,$$

we have, with probability no smaller than $1 - 14d^{-1}$, $\operatorname{sign}(A_1) = \operatorname{sign}(\breve{A}_1)$*. Here for any matrix* M, $\operatorname{sign}(M)$ *is a matrix with each element representing the sign of the corresponding entry in* M*.*

Here we note that Corollary 1 sheds lights to detecting Granger causality. For any two processes $\{y_t\}$ and $\{z_t\}$, Granger (1969) defined the causal relationship in principle as follows: Provided that we know everything in the universe, $\{y_t\}$ is said to cause $\{z_t\}$ in Granger's sense if removing the information about $\{y_s\}_{s\leq t}$ from the whole knowledge base built by time t will increase the prediction error about z_t. It is known that the noncausalities are determined by the transition matrices in the stable VAR process (Lütkepohl, 2005). Therefore, detecting the nonzero entries of A_1 consistently means that we can estimate the Granger-causality network consistently.

We then turn to evaluate the prediction performance of the proposed method. Given a new data point X_{T+1} in the time point $T+1$, based on $(X_t)_{t=1}^T$, the next corollary quantifies the distance between X_{T+1} and $\widehat{A}_1 X_T$ in terms of L_∞ norm.

Corollary 2 *Suppose that the conditions in Theorem 1 hold and let*

$$\Psi_{\max} := \max_i(\Psi_{ii}) \quad \text{and} \quad \Sigma_{\max} := \max_i(\Sigma_{ii}).$$

Then for the new data point X_{T+1} at time point $T+1$ and any constant $\alpha > 0$, with probability greater than

$$1 - 2(d^{\alpha/2-1}\sqrt{\pi/2 \cdot \alpha \log d}\})^{-1} - 14d^{-1},$$

we have

$$\|X_{T+1} - \widehat{A}_1^{\mathrm{T}} X_T\|_\infty \leq (\Psi_{\max} \cdot \alpha \log d)^{1/2} + 4s \left\{ \frac{32\|\Sigma^{-1}\|_1 \max_j(\Sigma_{jj})\|\Sigma\|_2}{\min_j(\Sigma_{jj})(1 - \|A_1\|_2)} (2M_d + 3) \left(\frac{\log d}{T}\right)^{1/2} \right\}^{1-q} \cdot (\Sigma_{\max} \cdot \alpha \log d)^{1/2}, \quad (20)$$

where $\widehat{A}_1$ is calculated based on $(X_t)_{t=1}^T$.

Here we note that the first term in the right-hand side of Equation (20), $(\Psi_{\max} \cdot \alpha \log d)^{1/2}$, is present due to the diverges of the new data point from its mean caused by an unpredictable noise perturb term $Z_{T+1} \sim N_d(0, \Psi)$. This term is unable to be canceled out even if we have almost infinite data points. The second term in the right-hand side of Equation (20) depends on the estimation accuracy of $\widehat{A}_1$ to A_1 and will converge to zero under certain conditions. In other words, the term

$$\|A_1^{\mathrm{T}} X_T - \widehat{A}_1^{\mathrm{T}} X_T\|_\infty \to 0,$$

converges to zero in probability as $n, d \to \infty$.

Although A_1 is in general asymmetric, there exist cases such that a symmetric transition matrix is more of interest. It is known that the off-diagonal entries in the transition matrix represent the influence of one state on the others and such influence might be symmetric or not. Weiner et al. (2012) provided several examples where a symmetric transition matrix is more appropriate for modeling the data.

If we can further suppose that the transition matrix A_1 is symmetric, we can use this information and obtain a new estimator $\bar{A}_1$ as

$$(\bar{A}_1)_{jk} = (\bar{A}_1)_{kj} := (\widehat{A}_1)_{jk} I(|(\widehat{A}_1)_{jk}| \leq |(\widehat{A}_1)_{kj}|) + (\widehat{A}_1)_{kj} I(|(\widehat{A}_1)_{kj}| \leq |(\widehat{A}_1)_{jk}|).$$

In other word, we always pick the entry with smaller magnitudes. Then using Theorem 1, we have $\|\bar{A}_1 - A_1\|_1$ and $\|\bar{A}_1 - A_1\|_\infty$ can be upper bounded by the same number presented in the right-hand side of (15). In this case, because both A_1 and $\bar{A}_1$ are symmetric, we have $\|\bar{A}_1 - A_1\|_2 \leq \|\bar{A}_1 - A_1\|_1 = \|\bar{A}_1 - A_1\|_\infty$. We then proceed to quantify the prediction accuracy under L_2 norm in the next corollary.

Corollary 3 *Suppose that the conditions in Theorem 1 hold and A_1 is a symmetric matrix. Then for the new data point X_{T+1} at time point $T+1$, with probability greater than $1-18d^{-1}$, we have*

$$\|X_{T+1} - \bar{A}_1^{\mathrm{T}} X_T\|_2 \leq \sqrt{2\|\Psi\|_2 \log d} + \sqrt{\mathrm{tr}(\Psi)} \; + 4s \left\{ \frac{32\|\Sigma^{-1}\|_1 \max_j(\Sigma_{jj})\|\Sigma\|_2}{\min_j(\Sigma_{jj})(1 - \|A_1\|_2)} (2M_d+3) \left(\frac{\log d}{T}\right)^{1/2} \right\}^{1-q} \cdot \{\sqrt{2\|\Sigma\|_2 \log d} + \sqrt{\mathrm{tr}(\Sigma)}\}. \quad (21)$$

Based on Corollary 3, we have, similar as what is discussed in Corollary 2, the term $\|A_1^{\mathrm{T}} X_T - \widehat{A}_1^{\mathrm{T}} X_T\|_2$ will vanish when the second term in the left-hand side of (21) can converge to zero.

5. Experiments

We conduct numerical experiments on both synthetic and real data to illustrate the effectiveness of our proposed method compared to the competing ones, as well as obtain more insights on the performance of the proposed method. In the following we consider the three competing methods:

- (i) The least square estimation using a ridge-penalty (The method in Hamilton, 1994, by adding a ridge-penalty $\|M\|_{\mathsf{F}}^2$ to the least squares loss function in Equation 1).
- (ii) The least square estimation using an L_1 penalty (The method in Hsu et al., 2008, by adding an L_1 penalty $\sum_{ij}|M_{ij}|$ to Equation 1).
- (iii) Our method (The estimator described in Equation 10).

Here we consider including the procedure discussed in Hamilton (1994) because it is a commonly explored baseline and shows how bad the classic procedure can be when the dimension is high. We only consider the competing procedure proposed in Hsu et al. (2008) because this is the only method that is specifically designed for the same simple VAR as what we study. We do not consider other aforementioned procedures (e.g., Haufe et al., 2008; Shojaie and Michailidis, 2010) because they are designed for more specific models with more assumptions. We use the R package "glmnet" (Friedman et al., 2010) for implementing the lasso method in Hsu et al. (2008), and the simplex algorithm for implementing ours.

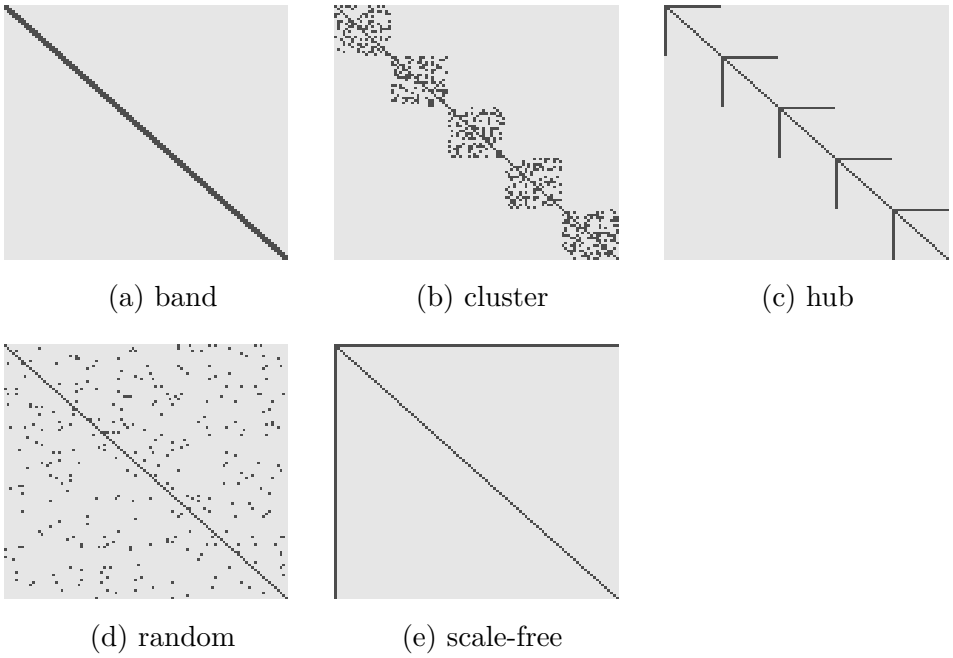
5.1 Cross-Validation Procedure

We start with an introduction to how to conduct cross-validation for choosing the lag p and the tuning parameter λ in the algorithm outlined in Section 3.

For the time series $(X_t)_{t=-\infty}^{T}$ and a specific time point t_0 of interest, if both p and λ are assumed to be unknown, the proposed cross-validation procedure is as follows.

1. We set all possible choices of (p,λ) to be a grid. We set n_1 and n_2 to be two numbers (representing the length of training data and the number of replicates).
2. For each X_t among $X_{t_0-1},\ldots,X_{t_0-n_2}$, the estimates $\widehat{A}_1^t(p,\lambda),\ldots,\widehat{A}_p^t(p,\lambda)$ are calculated based on the training data $X_{t-1},\ldots,X_{t-n_1}$ and any choice of (p,λ). We set the prediction error at time t, denoted as $\mathrm{Err}_t(p,\lambda)$, to be $\mathrm{Err}_t(p,\lambda) := \|X_t - \sum_{k=1}^{p}\widehat{A}_k^t(p,\lambda)^{\mathrm{T}}X_{t-k}\|_2$.
3. We take an average over the prediction errors and denote
$$\overline{\mathrm{Err}}(p,\lambda) := \frac{1}{n_2}\sum_{t=t_0-n_2}^{t_0-1}\mathrm{Err}_t(p,\lambda)$$
.
4. We choose the (p,λ) over the grid such that $\overline{\mathrm{Err}}(p,\lambda)$ is minimized.

In case when p is predetermined, the above procedure can be easily modified to focus only on selecting λ with p to be the determined value.



(a) band (b) cluster (c) hub

(d) random (e) scale-free

Figure 1: Five different transition matrix patterns used in the experiments. Here gray points represent the zero entries and black points represent nonzero entries.

5.2 Synthetic Data Analysis

In this subsection, we compare the performance of our method with the ridge and lasso methods using synthetic data under multiple settings. We also study the impact of transition matrices' spectral norms on estimation accuracy, and how the computation time and memory usage of all methods scale with the number of lags.

5.2.1 Performance Comparison: Lag $p = 1$

This section focuses on vector autoregressive model described in (2) with lag one. We compare our method to the competing ones on several synthetic data sets. We consider the settings where the time series length T varies from 50 to 100 and the dimension d varies from 50 to 200.

We create the transition matrix A_1 according to five different patterns: band, cluster, hub, random, and scale-free. Typical realizations of these patterns are illustrated in Figure 1 and are generated using the "flare" package in R (Li et al., 2015). In those plots, the gray points represent the zero entries and the black points represent the nonzero entries. We then rescale A_1 such that we have $\|A_1\|_2 = 0.5$. Once A_1 is obtained, we generate Σ using two models. First is the simple setting with Σ to be diagonal

$$\Sigma = 2\|A_1\|_2 I_d. \tag{22}$$

		ridge method			lasso method			our method		
d	T	L_{F}	L_2	L_1	L_{F}	L_2	L_1	L_{F}	L_2	L_1
50	100	2.71	0.52	2.47	2.34	0.50	1.54	**2.08**	**0.49**	**0.58**
		(0.028)	(0.023)	(0.103)	(0.064)	(0.029)	(0.161)	(0.045)	(0.006)	(0.039)
100	50	4.21	0.64	3.54	5.52	0.75	3.13	**3.26**	**0.52**	**1.03**
		(0.026)	(0.024)	(0.136)	(0.075)	(0.024)	(0.211)	(0.052)	(0.017)	(0.321)
200	100	7.28	0.76	6.26	6.36	0.64	2.77	**4.26**	**0.50**	**0.69**
		(0.031)	(0.018)	(0.132)	(0.057)	(0.015)	(0.112)	(0.045)	(0.003)	(0.035)

Table 1: Comparison of estimation performance of three methods with diagonal covariance matrix over 1,000 replications. The standard deviations are presented in the parentheses. Here L_{F}, L_2, and L_1 represent the Frobenius, L_2, and L_1 matrix norms respectively. The pattern of the transition matrix is "band".

The second is the complex setting where Σ is of Toeplitz form

$$\Sigma_{i,i} = 1, \quad \Sigma_{i,j} = \rho^{|i-j|} \text{ for some } \rho \in (0,1) \text{ and } i,j = 1,\ldots,d.$$

We then calculate the covariance matrix Ψ of the Gaussian noise vector Z_t as $\Psi = \Sigma - A_1^{\mathrm{T}}\Sigma A_1$. With A_1, Σ, and Ψ, we simulate a time series $(X_1,\ldots,X_T)^{\mathrm{T}} \in \mathbb{R}^{T\times d}$ according to the model described in (2).

We construct $1,000$ replicates and compare the three methods described above. The averaged estimation errors under different matrix norms are illustrated in Tables 1 to 10. The standard deviations of the estimation errors are provided in the parentheses. The tuning parameters for the three methods are selected using the cross-validation procedure outlined in Section 5.1 with $n_1 = T/2$, $n_2 = T/2$, and the lag p predetermined to be 1.

Tables 1 to 10 show that our method nearly uniformly outperforms the methods in Hsu et al. (2008) and Hamilton (1994) under different norms (Frobenius, L_2, and L_1 norms). In particular, the improvement over the method in Hsu et al. (2008) tends to be more significant when the dimension d is larger. Our method also has averagely slightly less standard deviations compared to the method in Hsu et al. (2008), but overall the difference is not significant. The method in Hamilton (1994) has worse performance than the other two methods. This verifies that it is not appropriate to handle very high dimensional data.

5.2.2 Synthetic Data: Lag $p \geq 1$

In this section, we further compare the performance of the three competing methods under the settings of possibly multiple lags, with the number of lags known.

In detail, we choose p to be from 1 to 9, the time series length $T = 100$, and the dimension $d = 50$. The transition matrices $A_1,\ldots,A_p$ are created according to "hub" or "scale-free" pattern, and then rescaled such that $\|A_i\|_2 = 0.1$ for $i = 1,\ldots,p$. The error covariance matrix Ψ is set to be identity for simplicity. Under this multiple lags setting, we then calculate the covariance matrix of $\widetilde{X}_t$, i.e., $\widetilde{\Sigma}$ defined in (5), by solving a discrete Lyapunov

		ridge method			lasso method			our method		
d	T	L_{F}	L_2	L_1	L_{F}	L_2	L_1	L_{F}	L_2	L_1
50	100	2.48	0.44	2.40	2.12	**0.43**	1.56	**1.48**	0.49	**0.69**
		(0.034)	(0.024)	(0.110)	(0.055)	(0.032)	(0.119)	(0.020)	(0.011)	(0.026)
100	50	3.74	0.58	3.46	5.24	0.67	3.16	**2.27**	**0.50**	**0.66**
		(0.031)	(0.022)	(0.121)	(0.084)	(0.025)	(0.223)	(0.002)	(0.001)	(0.002)
200	100	6.80	0.72	6.26	5.82	0.55	2.80	**3.02**	**0.49**	**0.77**
		(0.025)	(0.021)	(0.188)	(0.058)	(0.014)	(0.109)	(0.024)	(0.010)	(0.047)

Table 2: Comparison of estimation performance of three methods with diagonal covariance matrix over 1,000 replications. The standard deviations are presented in the parentheses. Here L_{F}, L_2, and L_1 represent the Frobenius, L_2, and L_1 matrix norms respectively. The pattern of the transition matrix is "cluster".

		ridge method			lasso method			our method		
d	T	L_{F}	L_2	L_1	L_{F}	L_2	L_1	L_{F}	L_2	L_1
50	100	2.41	0.42	2.37	1.96	**0.38**	1.48	**1.16**	0.41	**1.05**
		(0.033)	(0.027)	(0.102)	(0.06)	(0.039)	(0.141)	(0.115)	(0.058)	(0.092)
100	50	3.49	0.55	3.44	5.06	0.63	3.11	**1.86**	**0.50**	**1.40**
		(0.034)	(0.023)	(0.143)	(0.088)	(0.032)	(0.214)	(0.118)	(0.016)	(0.138)
200	100	6.61	0.69	6.24	5.48	0.52	2.75	**2.12**	**0.50**	**1.26**
		(0.035)	(0.017)	(0.133)	(0.062)	(0.019)	(0.147)	(0.046)	(0.006)	(0.031)

Table 3: Comparison of estimation performance of three methods with diagonal covariance matrix over 1,000 replications. The standard deviations are presented in the parentheses. Here L_{F}, L_2, and L_1 represent the Frobenius, L_2, and L_1 matrix norms respectively. The pattern of the transition matrix is "hub".

		ridge method			lasso method			our method		
d	T	L_{F}	L_2	L_1	L_{F}	L_2	L_1	L_{F}	L_2	L_1
50	100	2.60	0.48	2.45	2.21	**0.43**	1.53	**1.73**	0.44	**0.73**
		(0.031)	(0.027)	(0.102)	(0.061)	(0.030)	(0.143)	(0.051)	(0.026)	(0.034)
100	50	4.10	0.61	3.53	5.44	0.71	3.09	**3.07**	**0.48**	**1.21**
		(0.025)	(0.020)	(0.136)	(0.077)	(0.024)	(0.224)	(0.066)	(0.024)	(0.177)
200	100	7.01	0.74	6.27	6.03	0.58	2.79	**3.54**	**0.44**	**0.95**
		(0.024)	(0.019)	(0.179)	(0.048)	(0.011)	(0.163)	(0.036)	(0.026)	(0.079)

Table 4: Comparison of estimation performance of three methods with diagonal covariance matrix over 1,000 replications. The standard deviations are presented in the parentheses. Here L_{F}, L_2, and L_1 represent the Frobenius, L_2, and L_1 matrix norms respectively. The pattern of the transition matrix is "random".

		ridge method			lasso method			our method		
d	T	L_{F}	L_2	L_1	L_{F}	L_2	L_1	L_{F}	L_2	L_1
50	100	2.48	0.44	2.40	2.09	0.41	1.51	**1.44**	**0.41**	**0.98**
		(0.032)	(0.025)	(0.098)	(0.059)	(0.033)	(0.154)	(0.075)	(0.052)	(0.108)
100	50	3.60	0.56	3.43	5.14	0.64	3.11	**2.16**	**0.46**	**1.36**
		(0.034)	(0.023)	(0.133)	(0.085)	(0.031)	(0.188)	(0.130)	(0.043)	(0.115)
200	100	6.65	0.70	6.26	5.57	0.51	3.29	**2.51**	**0.42**	**2.49**
		(0.034)	(0.017)	(0.143)	(0.065)	(0.014)	(0.274)	(0.249)	(0.050)	(0.108)

Table 5: Comparison of estimation performance of three methods with diagonal covariance matrix over 1,000 replications. The standard deviations are presented in the parentheses. Here L_{F}, L_2, and L_1 represent the Frobenius, L_2, and L_1 matrix norms respectively. The pattern of the transition matrix is "scale-free".

		ridge method			lasso method			our method		
d	T	L_{F}	L_2	L_1	L_{F}	L_2	L_1	L_{F}	L_2	L_1
50	100	2.47	0.51	2.25	2.10	**0.45**	1.32	**1.82**	0.47	**0.57**
		(0.031)	(0.033)	(0.101)	(0.066)	(0.035)	(0.131)	(0.084)	(0.014)	(0.044)
100	50	3.98	0.67	3.31	5.22	0.74	2.81	**3.15**	**0.51**	**1.04**
		(0.029)	(0.033)	(0.107)	(0.083)	(0.032)	(0.174)	(0.114)	(0.063)	(0.529)
200	100	6.92	0.79	5.96	5.82	0.61	2.44	**3.79**	**0.48**	**0.67**
		(0.033)	(0.028)	(0.142)	(0.060)	(0.023)	(0.134)	(0.078)	(0.006)	(0.034)

Table 6: Comparison of estimation performance of three methods on data generated with Toeplitz covariance matrix ($\rho = 0.5$), over 1,000 replications. The standard deviations are presented in the parentheses. Here L_{F}, L_2, and L_1 represent the Frobenius, L_2, and L_1 matrix norms respectively. The pattern of the transition matrix is "band".

		ridge method			lasso method			our method		
d	T	L_{F}	L_2	L_1	L_{F}	L_2	L_1	L_{F}	L_2	L_1
50	100	2.32	0.42	2.25	2.01	**0.39**	1.42	**1.46**	0.47	**0.69**
		(0.041)	(0.029)	(0.114)	(0.066)	(0.030)	(0.124)	(0.027)	(0.019)	(0.037)
100	50	3.61	0.57	3.33	5.08	0.65	3.01	**2.47**	**0.47**	**1.02**
		(0.034)	(0.029)	(0.124)	(0.087)	(0.031)	(0.212)	(0.075)	(0.031)	(0.155)
200	100	6.63	0.70	6.13	5.58	0.54	2.59	**2.96**	**0.48**	**0.79**
		(0.038)	(0.020)	(0.162)	(0.069)	(0.019)	(0.153)	(0.027)	(0.013)	(0.046)

Table 7: Comparison of estimation performance of three methods on data generated with Toeplitz covariance matrix ($\rho = 0.5$), over 1,000 replications. The standard deviations are presented in the parentheses. Here L_{F}, L_2, and L_1 represent the Frobenius, L_2, and L_1 matrix norms respectively. The pattern of the transition matrix is "cluster".

		ridge method			lasso method			our method		
d	T	L_{F}	L_2	L_1	L_{F}	L_2	L_1	L_{F}	L_2	L_1
50	100	2.27	0.40	2.22	1.85	**0.36**	1.34	**1.16**	0.39	**1.01**
		(0.039)	(0.037)	(0.099)	(0.067)	(0.041)	(0.157)	(0.124)	(0.062)	(0.102)
100	50	3.37	0.54	3.26	4.94	0.61	2.96	**1.86**	**0.50**	**1.37**
		(0.041)	(0.034)	(0.125)	(0.102)	(0.033)	(0.222)	(0.120)	(0.017)	(0.104)
200	100	6.46	0.67	6.19	5.24	0.50	2.54	**2.13**	**0.49**	**1.24**
		(0.042)	(0.024)	(0.168)	(0.071)	(0.025)	(0.162)	(0.107)	(0.023)	(0.042)

Table 8: Comparison of estimation performance of three methods on data generated with Toeplitz covariance matrix ($\rho = 0.5$), over 1,000 replications. The standard deviations are presented in the parentheses. Here L_{F}, L_2, and L_1 represent the Frobenius, L_2, and L_1 matrix norms respectively. The pattern of the transition matrix is "hub".

		ridge method			lasso method			our method		
d	T	L_{F}	L_2	L_1	L_{F}	L_2	L_1	L_{F}	L_2	L_1
50	100	2.49	0.45	2.34	2.15	**0.41**	1.44	**1.74**	0.44	**0.74**
		(0.036)	(0.029)	(0.104)	(0.071)	(0.032)	(0.139)	(0.058)	(0.033)	(0.043)
100	50	4.02	0.60	3.42	5.34	0.70	2.96	**3.07**	**0.47**	**1.21**
		(0.029)	(0.024)	(0.123)	(0.092)	(0.028)	(0.207)	(0.085)	(0.027)	(0.192)
200	100	6.89	0.72	6.13	5.87	0.56	2.65	**3.54**	**0.43**	**0.97**
		(0.028)	(0.022)	(0.164)	(0.057)	(0.016)	(0.174)	(0.052)	(0.019)	(0.091)

Table 9: Comparison of estimation performance of three methods on data generated with Toeplitz covariance matrix ($\rho = 0.5$), over 1,000 replications. The standard deviations are presented in the parentheses. Here L_{F}, L_2, and L_1 represent the Frobenius, L_2, and L_1 matrix norms respectively. The pattern of the transition matrix is "random".

		ridge method			lasso method			our method		
d	T	L_{F}	L_2	L_1	L_{F}	L_2	L_1	L_{F}	L_2	L_1
50	100	2.36	0.42	2.27	2.00	0.38	1.36	**1.42**	**0.37**	**0.89**
		(0.036)	(0.033)	(0.094)	(0.064)	(0.033)	(0.136)	(0.068)	(0.056)	(0.108)
100	50	3.49	0.55	3.29	5.03	0.63	2.96	**2.21**	**0.42**	**1.29**
		(0.039)	(0.029)	(0.124)	(0.100)	(0.027)	(0.212)	(0.149)	(0.050)	(0.131)
200	100	6.52	0.67	6.18	5.36	0.49	3.06	**2.55**	**0.39**	**2.44**
		(0.041)	(0.019)	(0.165)	(0.070)	(0.013)	(0.219)	(0.364)	(0.062)	(0.134)

Table 10: Comparison of estimation performance of three methods on data generated with Toeplitz covariance matrix ($\rho = 0.5$), over 1,000 replications. The standard deviations are presented in the parentheses. Here L_{F}, L_2, and L_1 represent the Frobenius, L_2, and L_1 matrix norms respectively. The pattern of the transition matrix is "scale-free".

equation $\widetilde{A}^{\mathrm{T}}\widetilde{\Sigma}\widetilde{A} - \widetilde{\Sigma} + \widetilde{\Psi} = 0$. This is via using the Matlab command "dlyapchol". With $\{A_i\}_{i=1}^p, \widetilde{\Sigma}$, and Ψ determined, we simulate a time series $(X_1, \ldots, X_T)^{\mathrm{T}} \in \mathbb{R}^{T\times d}$ according to the model described in (2) (with lag $p \geq 1$).

The estimation error is calculated by measuring the difference of $(A_1^{\mathrm{T}}, \ldots, A_p^{\mathrm{T}})^{\mathrm{T}}$ and $(\widehat{A}_1^{\mathrm{T}}, \ldots, \widehat{A}_p^{\mathrm{T}})^{\mathrm{T}}$ with regard to different matrix norms (L_{F}, L_2, and L_1 norms). We conduct $1,000$ simulations and compare the averaged performance of three competing methods. The calculated averaged estimation errors are illustrated in Tables 11 and 12. The standard deviations of the estimation errors are provided in the parentheses. Here the tuning parameters are selected in the same way as before. Tables 11 and 12 confirms that our method still outperforms the competing two methods.

5.2.3 Synthetic Data: Impact of Transition Matrices' Spectral Norms

In this section we illustrate the effects of the transition matrices' spectral norms on estimation accuracy. To this end, we study the settings in Section 5.2. More specifically, we set lag $p = 1$, the dimension d and the sample size T to be $d = 50$ and $T = 100$. The transition matrix A_1 is created according to different patterns ("band", "cluster", "hub", "scale-free", and "random"), and then rescaled such that $\|A_1\|_2 = \kappa$, where κ is from 0.05 to 0.9. Covariance matrix Σ is set to be of the form (22), and Ψ is accordingly determined by stationary condition. We select the tuning parameters using the cross-validation procedure as before. The estimation errors are then plotted against κ and shown in Figure 2.

Figure 2 illustrates that the estimation error is an increasing function of the spectral norm $\|A_1\|_2$. This demonstrates that the spectral norms of the transition matrices play an important role in estimation accuracy and justifies the theorems in Section 4.

	ridge method			lasso method			our method		
p	L_{F}	L_2	L_1	L_{F}	L_2	L_1	L_{F}	L_2	L_1
1	6.93	2.50	7.35	1.83	0.52	1.36	**0.25**	**0.11**	**0.23**
	(0.012)	(0.094)	(0.377)	(0.039)	(0.017)	(0.128)	(0.014)	(0.016)	(0.002)
3	9.13	2.89	15.96	2.52	0.59	2.18	**0.45**	**0.18**	**0.70**
	(0.129)	(0.092)	(0.249)	(0.085)	(0.016)	(0.116)	(0.023)	(0.004)	(0.003)
5	5.57	1.57	11.73	2.75	0.61	3.19	**0.58**	**0.23**	**1.23**
	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)
7	4.27	1.14	10.92	2.90	0.60	3.44	**0.72**	**0.31**	**1.83**
	(0.010)	(0.041)	(0.152)	(0.026)	(0.025)	(0.183)	(0.077)	(0.067)	(0.222)
9	3.59	0.90	10.17	2.98	0.61	4.11	**0.70**	**0.30**	**2.11**
	(0.026)	(0.023)	(0.219)	(0.061)	(0.004)	(0.201)	(0.000)	(0.000)	(0.000)

Table 11: Comparison of estimation performance of three methods over 1,000 replications under multiple lag settings. The standard deviations are presented in the parentheses. Here L_{F}, L_2, and L_1 represent the Frobenius, L_2, and L_1 matrix norms respectively. The pattern of the transition matrix is "hub".

	ridge method			lasso method			our method		
p	L_{F}	L_2	L_1	L_{F}	L_2	L_1	L_{F}	L_2	L_1
1	6.93	2.51	7.39	1.83	0.53	1.35	**0.30**	**0.12**	**0.24**
	(0.116)	(0.093)	(0.340)	(0.041)	(0.018)	(0.129)	(0.045)	(0.016)	(0.039)
3	9.14	3.00	15.97	2.53	0.60	2.19	**0.46**	**0.17**	**0.57**
	(0.133)	(0.099)	(0.219)	(0.090)	(0.020)	(0.094)	(0.058)	(0.007)	(0.083)
5	5.58	1.57	11.66	2.77	0.60	2.97	**0.62**	**0.23**	**0.93**
	(0.002)	(0.002)	(0.018)	(0.001)	(0.002)	(0.076)	(0.012)	(0.002)	(0.078)
7	4.28	1.14	10.97	2.90	0.60	3.34	**0.69**	**0.24**	**1.29**
	(0.014)	(0.042)	(0.164)	(0.031)	(0.020)	(0.131)	(0.041)	(0.005)	(0.078)
9	3.62	0.90	10.25	3.01	0.61	3.42	**0.87**	**0.30**	**1.79**
	(0.024)	(0.023)	(0.267)	(0.058)	(0.003)	(0.112)	(0.078)	(0.012)	(0.198)

Table 12: Comparison of estimation performance of three methods over 1,000 replications under multiple lag settings. The standard deviations are presented in the parentheses. Here L_{F}, L_2, and L_1 represent the Frobenius, L_2, and L_1 matrix norms respectively. The pattern of the transition matrix is "scale-free".

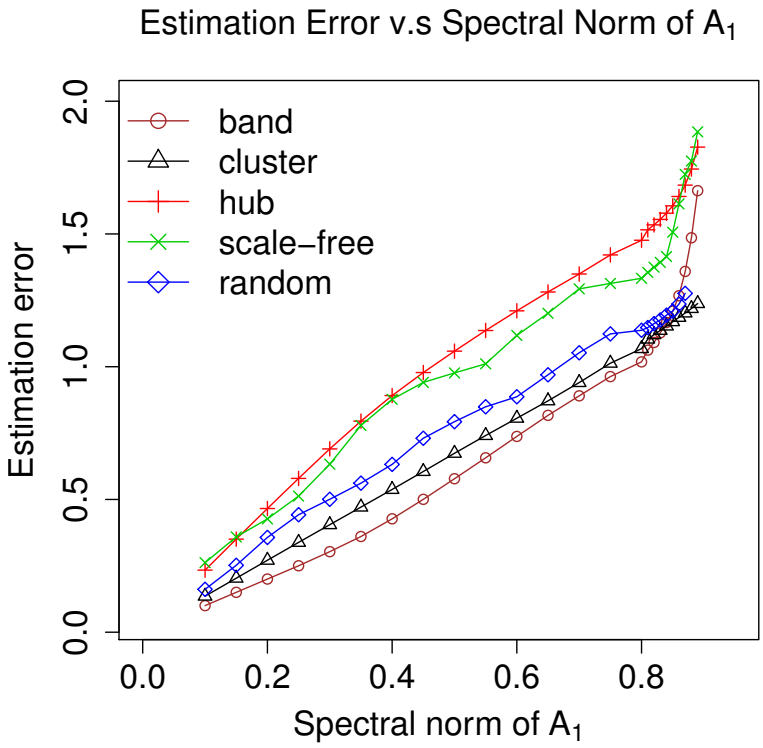


Figure 2: Estimation errors of A_1 (in L_1 norm) plotted against spectral norms of A_1.

5.2.4 Computation Time and Memory Usage

This section is devoted to show the computation time and memory usage of our method. First, we show the advantage of our method in terms of saving the computation time. A major advantage of our method over the two competing methods is that our method can be easily parallelly computed, and hence has the potential to save the computation time. We illustrate this point with a figure and two tables using the computation time as a function of the number of available cores. All experiments are conducted on a 2816-core Westmere/Ivybridge 2.67/2.5GHz Linux server with 17T memory, a cluster system with batch scheduling.

We first focus on the lag $p = 1$ case. In detail, we set the time series length $T = 100$ and the dimension $d = 50$. The transition matrix A_1 is created according to the pattern "random", and then rescaled such that $\|A_1\|_2 = 0.5$. The covariance matrix Σ is generated as in (22), and Ψ is generated by stationary condition. We then solve (11) using parallel computing based on 1 to 50 cores.

Figure 3 shows the computation time. It illustrates that, in terms of saving the computation time, under this specific setting, we have: (i) Our method outperforms the ridge method even if we do not parallelly compute it; (ii) When there are no less than 8 cores, our method outperforms the lasso method. Here the ridge method is very slow because it involves calculating the inverse of a large matrix.

To further study the advantage of parallel computing when the number of lags, p, grows, we provide another experiment focusing on models with varying lags. More specifically, we

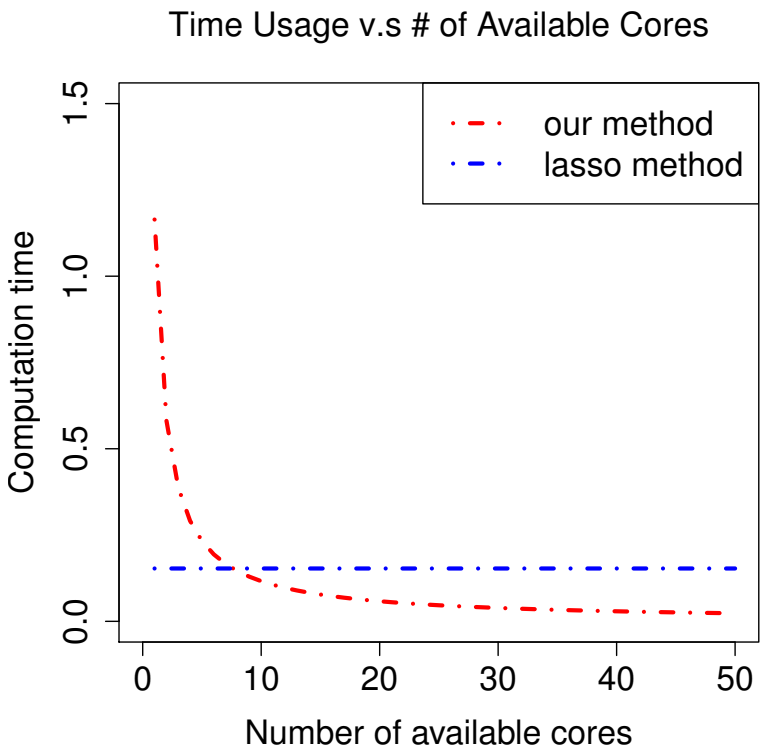


Figure 3: Computation time v.s number of available cores. The computation time for the ridge and lasso methods are 5.843s and 0.153s, which do not change with number of available cores. The computation time here is the averaged elapsed time (in seconds) of 100 replicates of a single experiment.

	lasso method	our method (with # of available cores)						
p	N/A	1	5	10	20	30	40	50
1	0.260	1.277	0.261	0.132	0.067	0.048	0.033	0.029
2	0.664	2.732	0.553	0.280	0.141	0.098	0.073	0.059
3	1.034	8.945	1.792	0.897	0.455	0.299	0.230	0.181
4	1.538	18.278	3.695	1.844	0.920	0.620	0.466	0.366
5	1.946	35.609	7.130	3.890	1.781	1.189	0.870	0.719

Table 13: A comparison of computation time with increasing number of lags p: lasso method v.s our method. The computation time for the lasso method does not change with number of available cores. The computation time here is the averaged elapsed time (in seconds) of 100 replicates of a single experiment.

set the time series length $T = 100$ and the dimension $d = 50$. The transition matrices $A_1, \ldots, A_p$ are created according to the pattern "random", and then rescaled such that $\|A_i\|_2 = 0.1$ for $i = 1, \ldots, p$. The error covariance matrix Ψ and the covariance matrix $\widetilde{\Sigma}$ are generated in the same way as in Section 5.2.2. With $\{A_i\}_{i=1}^p, \widetilde{\Sigma}$, and Ψ determined, we simulate a time series $(X_1, \ldots, X_T)^{\mathrm{T}} \in \mathbb{R}^{T \times d}$ according to the model described in (2). We then solve (11) using parallel computing based on 1 to 50 cores.

Table 13 lists the averaged elapsed time of 100 replicates of one single experiment. Here for each replication, the parameters $(A_1, \ldots, A_p, \Psi, \widetilde{\Sigma})$ in the experiment are regenerated. It illustrates that, in terms of saving the computation time, under this specific setting, we have: (i) When there is only one core, the lasso method outperforms our method. But when there are no less than 20 cores, our method outperforms the lasso method for all lags $p = 1, 2, 3, 4, 5$; (ii) As p grows, the advantage of parallel computing will be less significant (The ratio of computation time between our method at the maximum number of available cores and the lasso method tends to increase). We also observe from Table 13 that: (iii) The computation time of our method is approximately increasing quadratically with regard to the lag p, while the computation time of the lasso method is approximately increasing linearly with regard to the lag p.

Similarly, to study the advantage of parallel computing when the dimension d grows, we provide an experiment focusing on models with varying dimensions. We consider the settings where the length $T = 100$, the lag $p = 1$, and the dimension d varies from 10 to 200. The transition matrix A_1 is created according to the pattern "random", and then rescaled such that $\|A_1\|_2 = 0.5$. The covariance matrix Σ is generated as in (22), and Ψ is generated by stationary condition. We then solve (11) using parallel computing based on 1 to 200 cores.

Similar to Table 13, Table 14 lists the computation time. It illustrates that, in terms of saving the computation time, under this specific setting, we have: (i) When there is only one core, the lasso method outperforms our method. But when using the maximum number of cores (i.e., d cores), our method outperforms the lasso method for all lags $p = 1, 2, 3, 4, 5$; (ii)

	lasso method	our method (with # of available cores)						
d	N/A	1	5	10	20	50	100	200
10	0.022	0.074	0.015	0.008	N/A	N/A	N/A	N/A
20	0.048	0.265	0.055	0.027	0.014	N/A	N/A	N/A
50	0.153	1.164	0.234	0.120	0.061	0.027	N/A	N/A
100	0.468	6.354	1.281	0.649	0.318	0.131	0.067	N/A
200	2.320	21.503	4.304	2.157	1.111	0.448	0.219	0.108

Table 14: A comparison of computation time with increasing dimension d: lasso method v.s our method. The computation time for the lasso method does not change with number of available cores. The computation time here is the averaged elapsed time (in seconds) of 100 replicates of a single experiment.

As d grows, the advantage of parallel computing will be more significant (The ratio between our method at the maximum number of available cores and the lasso method decreases).

Tables 13 and 14 illustrate that, when p or d grows, the advantage of parallel computing becomes less or more significant respectively. Such results are reasonable because (3.4) can be decomposed to at most d subproblems in a columnwise way, and solved in parallel. As the dimension d grows, the maximum number of decomposed subproblems accordingly grows, and hence the gain in parallel computing will be more significant. In comparison, as p grows (while d is fixed), the maximum number of subproblems does not grow, and hence the advantage of parallel computing is less significant.

Secondly, we show the memory usage of our method. By converting the time series from VAR(1) to VAR(p) or increasing the dimension d, the memory usage increases. For investigating the memory usage, we conduct an empirical study. Specifically, first, we choose the lag p to be $1, 2, \ldots, 9$, the time series length $T = 100$, and the dimension $d = 50$. Transition matrices $A_1, \ldots, A_p$ are created according to the "random" pattern, and then rescaled such that $\|A_i\|_2 = 0.1$ for $i = 1, \ldots, p$. Ψ is set as I_d for simplicity. With $\{A_i\}_{i=1}^p$ and Ψ, we simulate a time series $(X_1, \ldots, X_T)^T \in \mathbb{R}^{T \times d}$ according to (2) with lag $p \geq 1$. The first two rows in Table 15 reports the averaged memory usage of 100 replicates of one single experiment in megabytes (Mb). Here for each replication, the parameters in the experiment are regenerated.

Secondly, we choose the lag $p = 1$, the time series length $T = 100$, the dimension $d = 50$, and the transition matrix A_1 to be created according to the "random" pattern, and then rescaled such that $\|A_1\|_2 = 0.1$. Ψ is set as I_d for simplicity. With A_1 and Ψ, we simulate a time series $(X_1, \ldots, X_T)^T \in \mathbb{R}^{T \times d}$ according to (2). The second two rows in Table 15 reports the memory usage.

Table 15 shows that, under this setting, the memory usage is approximately increasing linearly with regard to p; On the other hand, the memory usage is approximately increasing quadratically with regard to d, and this pattern becomes clearer when d is larger.

Lag of model (p)	1	2	3	4	5	6	7	8	9
Mem. Use (Mb)	5.566	8.724	11.862	14.999	18.135	21.272	24.406	27.540	30.673
Dimension (d)	10	20	30	40	50	75	100	150	200
Mem. Use (Mb)	1.649	2.235	3.083	4.194	5.566	10.150	16.390	33.754	57.678

Table 15: Memory usage v.s lag of model and dimension: The result shown below is the averaged memory usage (in Mb) of 100 replicates of one single experiment, with the lag p changing from 1 to 9 or dimension changing from 10 to 200. The pattern of the transition matrices $\{A_i\}_{i=1}^p$ is "random".

5.3 Real Data

We further compare the three methods on the equity data collected from Yahoo! Finance. The task is to predict the stock prices. We collect the daily closing prices for 91 stocks that are consistently in the S&P 100 index between January 1, 2003 and January 1, 2008. This gives us altogether 1,258 data points, each of which corresponds to the vector of closing prices on a trading day.

We first provide comparison on averaged prediction errors for using different lag p on this data set. Let $E = (E_{t,j}) \in \mathbb{R}^{1258\times 91}$ with $E_{t,j}$ denoting the closing price of the stock j on day t. We screen out all the stocks with low marginal standard deviations and only keep 50 stocks which vary the most. We center the data so that the marginal mean of each time series is zero. The resulting data matrix is denoted by $\bar{E} \in \mathbb{R}^{1258\times 50}$. We apply the three methods on $\bar{E}$ with different lag p changing from 1 to 9. To evaluate the performance of the three methods, for $t = 1248, \ldots, 1257$, we select the data set $\bar{E}_{J_t,*}$, where we have $J_t = \{j : t - 100 \le j \le t - 1\}$, as the training set. Then for each p and λ, based on the training set $\bar{E}_{J_t,*}$, we calculate the transition matrix estimates $\widehat{A}_1^t(p,\lambda), \ldots, \widehat{A}_p^t(p,\lambda)$. We then use the obtained estimates to predict the stock price in day t. The averaged prediction error for each specific λ and p is calculated as

$$\overline{\mathrm{Err}}(p,\lambda) = \frac{1}{10}\sum_{t=1}^{10} \|\bar{E}_{t,*} - \sum_{k=1}^{p} \widehat{A}_k^t(p,\lambda)^{\mathrm{T}} \bar{E}_{t-k,*}\|_2.$$

In Table 16, we present the minimized averaged prediction errors $\min_\lambda \overline{\mathrm{Err}}(p,\lambda)$ for the three methods with different lag p. The standard deviations of the prediction errors are presented in the parentheses. Our method outperforms the two competing methods in terms of prediction accuracy.

Secondly, we provide the prediction error on day $t = 1258$ based on the selected (p,λ) using cross-validation. By observing Table 16, we select the lag $p = 1$ and the corresponding λ for our method. The prediction error is 7.62 for our method. In comparison, the lasso method and ridge method have the prediction errors 11.11 and 11.94 separately.

lag	ridge method	lasso method	our method
p=1	17.68 (2.49)	15.67 (2.74)	**11.88** (3.34)
p=2	15.63 (3.01)	15.69 (2.84)	12.01 (3.41)
p=3	15.17 (3.53)	15.76 (2.83)	12.04 (3.42)
p=4	14.90 (3.69)	15.68 (2.76)	12.02 (3.41)
p=5	14.73 (3.66)	15.62 (2.55)	12.08 (3.29)
p=6	14.58 (3.57)	15.51 (2.58)	12.09 (3.15)
p=7	14.42 (3.49)	15.45 (2.59)	12.21 (3.16)
p=8	14.36 (3.42)	15.40 (2.57)	12.25 (3.16)
p=9	**14.20** (3.31)	**15.28** (2.46)	12.24 (3.06)

Table 16: The optimized averaged prediction errors for the three methods on the equity data, under different lags p from 1 to 9. The standard deviations are present in the parentheses. The smallest prediction error within each column is bolded.

6. Discussions

Estimation of the vector autoregressive model is an interesting problem and has been investigated for a long time. This problem is intrinsically linked to the regression problem with multiple responses. Accordingly (penalized) least squares estimates, which has the maximum likelihood interpretation behind it, look like reasonable solutions. However, high dimensionality brings significantly new challenges and viewpoints to this classic problem. In parallel to the Dantzig selector proposed by Candes and Tao (2007) in cracking the ordinary linear regression model, we advocate borrowing the strength of the linear program in estimating the VAR model. As has been repeatedly stated in the main text, this new formulation brings some advantages over the least square estimates. Moreover, our theoretical analysis brings new insights into the problem of transition matrix estimation, and we highlight the role of $\|A_1\|_2$ in evaluating the estimation accuracy of the estimator.

In the main text we do not discuss estimating the covariance matrix Σ and Ψ. Lemma 1 builds the $L_{\max}$ convergence result for estimating Σ. If we further suppose that the covariance matrix Σ is sparse in some sense, then we can exploit the well developed results in covariance matrix estimation (including "banding", Bickel and Levina, 2008b, "tapering", Cai et al., 2010, and "thresholding", Bickel and Levina, 2008a) to estimate the covariance matrix Σ and establish the consistency result with regard to the matrix L_1 and L_2 norms. With both Σ and A estimated by some constant estimator $\widehat{\Sigma}$, an estimator $\widehat{\Psi}$ of Ψ can be obtained under the VAR model (with lag one) as

$$\widehat{\Psi} = \widehat{\Sigma} - \widehat{A}_1^{\mathrm{T}} \widehat{\Sigma} \widehat{A}_1,$$

and a similar estimator can be built for lag p VAR model using the augmented formulation shown in Equation (4).

In this manuscript we focus on the stationary vector autoregressive model and our method is designed for such stationary process. The stationary requirement is a common

assumption in analysis and is adopted by most recent works, for example, Kock and Callot (2015) and Song and Bickel (2011). We notice that there are works in handling unstable VAR models, checking for example Song et al. (2014) and Kock (2012). We would like to explore this problem in the future. Another unexplored region is how to determine the order (lag) of the vector autoregression aside from using the cross-validation approach. There have been results in this area (e.g., Song and Bickel, 2011) and we are also interested in finding whether the linear program can also be exploited in determining the order of the VAR model.

Acknowledgments

We thank the associate editor and three anonymous referees for their helpful comments. Fang's research is supported by a Google fellowship. Han Liu is grateful for the support of NSF CAREER Award DMS1454377, NSF IIS1408910, NSF IIS1332109, NIH R01MH102339, NIH R01GM083084, and NIH R01HG06841.

Appendix A. Proofs of Main Results

In this section we provide the proofs of the main results in the manuscript.

A.1 Proof of Theorem 1

Before proving the main result in Theorem 1, we first establish several lemmas. In the sequel, because we only focus on the lag 1 autoregressive model, for notation simplicity, in $\Sigma_i(\{(X_t)\})$ we remove $\{(X_t)\}$ and simply denote the lag i covariance matrix to be Σ_i.

The following lemma describes the $L_{\max}$ rate of convergence S to Σ. This result generalizes the upper bound derived when data are independently generated (see, for example, Bickel and Levina, 2008a).

Lemma 1 *Letting S be the marginal sample covariance matrix defined in* (7)*, when $T \geq \max(6 \log d, 1)$, we have, with probability no smaller than $1 - 6d^{-1}$,*

$$\|S - \Sigma\|_{\max} \leq \frac{16\|\Sigma\|_2 \max_j(\Sigma_{jj})}{\min_j(\Sigma_{jj})(1 - \|A_1\|_2)} \left\{ \left(\frac{6 \log d}{T} \right)^{1/2} + 2 \left(\frac{1}{T} \right)^{1/2} \right\}.$$

Proof [**Proof**] For any $j, k \in \{1, 2, \ldots, d\}$, we have

$$\mathbb{P}(|S_{jk} - \Sigma_{jk}| > \eta) = \mathbb{P}\left(\left| \frac{1}{T} \sum_{t=1}^{T} X_{tj} \; X_{tk} - \Sigma_{jk} \right| > \eta \right).$$

Letting $Y_t = \{X_{t1}(\Sigma_{11})^{-1/2}, \ldots, X_{td}(\Sigma_{dd})^{-1/2}\}^{\mathrm{T}}$ for $t = 1, \ldots, T$ and $\rho_{jk} = \Sigma_{jk}(\Sigma_{jj}\Sigma_{kk})^{-1/2}$, we have

$$\begin{aligned}
\mathbb{P}(|S_{jk} - \Sigma_{jk}| > \eta) &= \mathbb{P}\left\{\left|\frac{1}{T}\sum_{t=1}^{T} Y_{tj}Y_{tk} - \rho_{jk}\right| > \eta(\Sigma_{jj}\Sigma_{kk})^{-1/2}\right\} \\
&= \mathbb{P}\left\{\left|\frac{\sum_{t=1}^{T}(Y_{tj}+Y_{tk})^2 - \sum_{t=1}^{T}(Y_{tj}-Y_{tk})^2}{4T} - \rho_{jk}\right| > \eta(\Sigma_{jj}\Sigma_{kk})^{-1/2}\right\} \\
&\le \mathbb{P}\left\{\left|\frac{1}{T}\sum_{t=1}^{T}(Y_{tj}+Y_{tk})^2 - 2(1+\rho_{jk})\right| > 2\eta(\Sigma_{jj}\Sigma_{kk})^{-1/2}\right\} \\
&\quad + \mathbb{P}\left\{\left|\frac{1}{T}\sum_{t=1}^{T}(Y_{tj}-Y_{tk})^2 - 2(1-\rho_{jk})\right| > 2\eta(\Sigma_{jj}\Sigma_{kk})^{-1/2}\right\}. \qquad (23)
\end{aligned}$$

Using the property of Gaussian distribution, we have $(Y_{1j}+Y_{1k}, \ldots, Y_{Tj}+Y_{Tk})^{\mathrm{T}} \sim N_T(0, Q)$ for some positive definite matrix Q. In particular, we have

$$\begin{aligned}
|Q_{il}| &= |\operatorname{Cov}(Y_{ij}+Y_{ik}, Y_{lj}+Y_{lk})| = |\operatorname{Cov}(Y_{ij}, Y_{lj}) + \operatorname{Cov}(Y_{ij}, Y_{lk}) + \operatorname{Cov}(Y_{ik}, Y_{lk}) + \operatorname{Cov}(Y_{ik}, Y_{lj})| \\
&\le \frac{1}{\min_j(\Sigma_{jj})}|\operatorname{Cov}(X_{ij}, X_{lj}) + \operatorname{Cov}(X_{ij}, X_{lk}) + \operatorname{Cov}(X_{ik}, X_{lk}) + \operatorname{Cov}(X_{ik}, X_{lj})| \\
&\le \frac{4}{\min_j(\Sigma_{jj})}\|\Sigma_{l-i}\|_{\max} \le \frac{8\|\Sigma\|_2\|A_1\|_2^{|l-i|}}{\min_j(\Sigma_{jj})},
\end{aligned}$$

where the last inequality follows from (3).

Therefore, using the matrix norm inequality,

$$\|Q\|_2 \le \max_{1\le i\le T}\sum_{l=1}^{T}|Q_{il}| \le \frac{8\|\Sigma\|_2}{\min_j(\Sigma_{jj})(1-\|A_1\|_2)}.$$

Then applying Lemma 3 to (23), we have

$$\begin{aligned}
&\mathbb{P}\left\{\left|\frac{1}{T}\sum_{t=1}^{T}(Y_{tj}+Y_{tk})^2 - 2(1+\rho_{jk})\right| > 2\eta(\Sigma_{jj}\Sigma_{kk})^{-1/2}\right\} \\
&\quad \le 2\exp\left[-\frac{T}{2}\left\{\frac{\eta\min_j(\Sigma_{jj})(1-\|A_1\|_2)}{16\|\Sigma\|_2(\Sigma_{jj}\Sigma_{kk})^{1/2}} - 2T^{-1/2}\right\}^2\right] + 2\exp\left(-\frac{T}{2}\right). \qquad (24)
\end{aligned}$$

Using a similar argument, we have

$$\begin{aligned}
&\mathbb{P}\left\{\left|\frac{1}{T}\sum_{t=1}^{T}(Y_{tj}-Y_{tk})^2 - 2(1-\rho_{jk})\right| > 2\eta(\Sigma_{jj}\Sigma_{kk})^{-1/2}\right\} \\
&\quad \le 2\exp\left[-\frac{T}{2}\left\{\frac{\eta\min_j(\Sigma_{jj})(1-\|A_1\|_2)}{16\|\Sigma\|_2(\Sigma_{jj}\Sigma_{kk})^{1/2}} - 2T^{-1/2}\right\}^2\right] + 2\exp\left(-\frac{T}{2}\right). \qquad (25)
\end{aligned}$$

Combining (24) and (25), then applying the union bound, we have

$$\mathbb{P}(\|S-\Sigma\|_{\max}>\eta) \leq 3d^2\exp\left(-\frac{T}{2}\right)+3d^2\exp\left[-\frac{T}{2}\left\{\frac{\eta\min_j(\Sigma_{jj})(1-\|A_1\|_2)}{16\|\Sigma\|_2\max_j(\Sigma_{jj})}-2\left(\frac{1}{T}\right)^{-1/2}\right\}^2\right].$$

The proof thus completes by choosing η as the described form. ■

In the next lemma we try to quantify the difference between S_1 and Σ_1 with respect to the matrix $L_{\max}$ norm. Remind that $\Sigma_1\{(X_t)\}$ is simplified to be Σ_1.

Lemma 2 *Letting S_1 be the lag 1 sample covariance matrix, when $T \geq \max(6\log d+1, 2)$, we have, with probability no smaller than $1-8d^{-1}$,*

$$\|S_1-\Sigma_1\|_{\max} \leq \frac{32\|\Sigma\|_2\max_j(\Sigma_{jj})}{\min_j(\Sigma_{jj})(1-\|A_1\|_2)}\left\{\left(\frac{3\log d}{T}\right)^{1/2}+\left(\frac{2}{T}\right)^{1/2}\right\}.$$

Proof [**Proof**] We have, for any $j,k\in\{1,2,\ldots,d\}$,

$$\mathbb{P}(|(S_1)_{jk}-(\Sigma_1)_{jk}|>\eta)=\mathbb{P}\left(\left|\frac{1}{T-1}\sum_{t=1}^{T-1}X_{tj}X_{(t+1)k}-(\Sigma_1)_{jk}\right|>\eta\right).$$

Letting $Y_t=\{X_{t1}(\Sigma_{11})^{-1/2},\ldots,X_{td}(\Sigma_{dd})^{-1/2}\}^{\mathrm{T}}$ and $\rho_{jk}=(\Sigma_1)_{jk}(\Sigma_{jj}\Sigma_{kk})^{-1/2}$, we have

$$\begin{aligned}
&\mathbb{P}(|(S_1)_{jk}-(\Sigma_1)_{jk}|>\eta)=\mathbb{P}\left\{\left|\frac{1}{T-1}\sum_{t=1}^{T-1}Y_{tj}Y_{(t+1)k}-\rho_{jk}\right|>\eta(\Sigma_{jj}\Sigma_{kk})^{-1/2}\right\}\\
&=\mathbb{P}\left[\left|\frac{\sum_{t=1}^{T-1}\{Y_{tj}+Y_{(t+1)k}\}^2-\sum_{t=1}^{T-1}\{Y_{tj}-Y_{(t+1)k}\}^2}{4(T-1)}-\rho_{jk}\right|>\eta(\Sigma_{jj}\Sigma_{kk})^{-1/2}\right]\\
&\leq\mathbb{P}\left[\left|\frac{\sum_{t=1}^{T-1}\{Y_{tj}+Y_{(t+1)k}\}^2}{T-1}-2(1+\rho_{jk})\right|>2\eta(\Sigma_{jj}\Sigma_{kk})^{-1/2}\right]\\
&\quad+\mathbb{P}\left[\left|\frac{\sum_{t=1}^{T-1}\{Y_{tj}-Y_{(t+1)k}\}^2}{T-1}-2(1-\rho_{jk})\right|>2\eta(\Sigma_{jj}\Sigma_{kk})^{-1/2}\right]. \qquad (26)
\end{aligned}$$

Using the property of Gaussian distribution, we have $\{Y_{1j}+Y_{2k},\ldots,Y_{(T-1)j}+Y_{Tk}\}^{\mathrm{T}}\sim N_{T-1}(0,Q)$, for some positive definite matrix Q. In particular, we have

$$\begin{aligned}
&|Q_{il}|=|\operatorname{Cov}\{Y_{ij}+Y_{(i+1)k},Y_{lj}+Y_{(l+1)k}\}|\\
&=|\operatorname{Cov}(Y_{ij},Y_{lj})+\operatorname{Cov}\{Y_{ij},Y_{(l+1)k}\}+\operatorname{Cov}\{Y_{(i+1)k},Y_{lj}\}+\operatorname{Cov}\{Y_{(i+1)k},Y_{(l+1)k}\}|\\
&\leq\frac{1}{\min_j(\Sigma_{jj})}|\operatorname{Cov}(X_{ij},X_{lj})+\operatorname{Cov}\{X_{ij},X_{(l+1)k}\}+\operatorname{Cov}\{X_{(i+1)k},X_{lj}\}+\operatorname{Cov}\{X_{(i+1)k},X_{(l+1)j}\}|\\
&\leq\frac{2\|\Sigma_{l-i}\|_{\max}+\|\Sigma_{l+1-i}\|_{\max}+\|\Sigma_{l-1-i}\|_{\max}}{\min_j(\Sigma_{jj})}\\
&\leq\frac{\|\Sigma\|_2(2\|A_1\|_2^{|l-i|}+\|A_1\|_2^{|l+1-i|}+\|A_1\|_2^{|l-1-i|})}{\min_j(\Sigma_{jj})}.
\end{aligned}$$

Therefore, using the matrix norm inequality,

$$\|Q\|_2 \leq \max_{1\leq i\leq (T-1)} \sum_{l=1}^{T-1} |Q_{il}| \leq \frac{8\|\Sigma\|_2}{\min_j(\Sigma_{jj})(1-\|A_1\|_2)}.$$

Then applying Lemma 3 to (26), we have

$$\mathbb{P}\left[\left|\frac{1}{T-1}\sum_{t=1}^{T-1}\{Y_{tj}+Y_{(t+1)k}\}^2 - 2(1+\rho_{jk})\right| > 2\eta(\Sigma_{jj}\Sigma_{kk})^{-1/2}\right] \leq$$
$$2\exp\left[-\frac{(T-1)}{2}\left\{\frac{\eta\min_j(\Sigma_{jj})(1-\|A_1\|_2)}{16\|\Sigma\|_2(\Sigma_{jj}\Sigma_{kk})^{1/2}} - 2(T-1)^{-1/2}\right\}^2\right] + 2\exp\left(-\frac{T-1}{2}\right). \quad (27)$$

Using a similar technique, we have

$$\mathbb{P}\left[\left|\frac{1}{T-1}\sum_{t=1}^{T-1}\{Y_{tj}-Y_{(t+1)k}\}^2 - 2(1-\rho_{jk})\right| > 2\eta(\Sigma_{jj}\Sigma_{kk})^{-1/2}\right] \leq$$
$$2\exp\left[-\frac{(T-1)}{2}\left\{\frac{\eta\min_j(\Sigma_{jj})(1-\|A_1\|_2)}{16\|\Sigma\|_2(\Sigma_{jj}\Sigma_{kk})^{1/2}} - 2(T-1)^{-1/2}\right\}^2\right] + 2\exp\left(-\frac{T-1}{2}\right). \quad (28)$$

Combining (27) and (28), and applying the union bound across all pairs (j,k), we have

$$\mathbb{P}(\|S_1-\Sigma_1\|_{\max} > \eta) \leq$$
$$4d^2\exp\left[-\frac{(T-1)}{2}\left\{\frac{\eta\min_j(\Sigma_{jj})(1-\|A_1\|_2)}{16\|\Sigma\|_2\max_j(\Sigma_{jj})} - 2(T-1)^{-1/2}\right\}^2\right] + 4d^2\exp\left(-\frac{T-1}{2}\right).$$

Finally noting that when $T \geq 3$, we have $1/(T-1) < 2/T$. The proof thus completes by choosing η as stated. ■

Using the above two technical lemmas, we can then proceed to the proof of the main results in Theorem 1.

Proof [**Proof of Theorem 1**] With Lemmas 1 and 2, we proceed to prove Theorem 1. We first denote

$$\zeta_1 = \frac{16\|\Sigma\|_2\max_j(\Sigma_{jj})}{\min_j(\Sigma_{jj})(1-\|A_1\|_2)}\left\{\left(\frac{6\log d}{T}\right)^{1/2} + 2\left(\frac{1}{T}\right)^{1/2}\right\},$$
$$\zeta_2 = \frac{32\|\Sigma\|_2\max_j(\Sigma_{jj})}{\min_j(\Sigma_{jj})(1-\|A_1\|)_2}\left\{\left(\frac{3\log d}{T}\right)^{1/2} + \left(\frac{2}{T}\right)^{1/2}\right\}.$$

Using Lemmas 1 and 2, we have, with probability no smaller than $1-14d^{-1}$,

$$\|S-\Sigma\|_{\max} \leq \zeta_1, \quad \|S_1-\Sigma_1\|_{\max} \leq \zeta_2.$$

We firstly prove that population quantity A_1 is a feasible solution to the optimization problem in (10) with probability no smaller than $1 - 14d^{-1}$

$$\begin{aligned}
\|SA_1 - S_1\|_{\max} &= \|S\Sigma^{-1}\Sigma_1 - S_1\|_{\max} \\
&= \|S\Sigma^{-1}\Sigma_1^{\mathrm{T}} - \Sigma_1 + \Sigma_1 - S_1\|_{\max} \\
&\le \|(S\Sigma^{-1} - I_d)\Sigma_1\|_{\max} + \|\Sigma_1 - S_1\|_{\max} \\
&\le \|(S - \Sigma)\Sigma^{-1}\Sigma_1\|_{\max} + \zeta_2 \\
&\le \zeta_1\|A_1\|_1 + \zeta_2 \\
&\le \lambda_0.
\end{aligned}$$

The last inequality holds by using the condition that $d \ge 8$ implies that $1/T \le \log d/(2T)$. Therefore, A_1 is feasible in the optimization equation, by checking the equivalence between (10) and (11), we have $\|\widehat{\Omega}\|_1 \le \|A_1\|_1$ with probability no smaller than $1 - 14d^{-1}$. We then have

$$\begin{aligned}
\|\widehat{\Omega} - A_1\|_{\max} &= \|\widehat{\Omega} - \Sigma^{-1}\Sigma_1\|_{\max} \\
&= \|\Sigma^{-1}(\Sigma\widehat{\Omega} - \Sigma_1)\|_{\max} \\
&= \|\Sigma^{-1}(\Sigma\widehat{\Omega} - S_1 + S_1 - \Sigma_1)\|_{\max} \\
&= \|\Sigma^{-1}(\Sigma\widehat{\Omega} - S\widehat{\Omega} + S\widehat{\Omega} - S_1) + \Sigma^{-1}(S_1 - \Sigma_1)\|_{\max} \\
&\le \|(I_d - \Sigma^{-1}S)\widehat{\Omega}\|_{\max} + \|\Sigma^{-1}(S\widehat{\Omega} - S_1)\|_{\max} + \|\Sigma^{-1}(S_1 - \Sigma_1)\|_{\max} \\
&\le \|\Sigma^{-1}\|_1\|(\Sigma - S)\widehat{\Omega}\|_{\max} + \|\Sigma^{-1}\|_1\|S\widehat{\Omega} - S_1\|_{\max} + \|\Sigma^{-1}\|_1\|S_1 - \Sigma_1\|_{\max} \\
&\le \|\Sigma^{-1}\|_1(\|A_1\|_1\zeta_1 + \lambda_0 + \zeta_2) \\
&= 2\lambda_0\|\Sigma^{-1}\|_1.
\end{aligned}$$

Let λ_1 be a threshold level and we define

$$s_1 = \max_{1\le j\le d}\sum_{i=1}^{d}\min\left\{|(A_1)_{ij}|/\lambda_1, 1\right\}, \qquad T_j = \left\{i : |(A_1)_{ij}| \ge \lambda_1\right\}.$$

We have, with probability no smaller than $1 - 14d^{-1}$, for all $j \in \{1, \ldots, d\}$,

$$\begin{aligned}
\|\widehat{\Omega}_{*,j} - (A_1)_{*,j}\|_1 &\le \|\widehat{\Omega}_{T_j^c,j}\|_1 + \|(A_1)_{T_j^c,j}\|_1 + \|\widehat{\Omega}_{T_j,j} - (A_1)_{T_j,j}\|_1 \\
&= \|\widehat{\Omega}_{*,j}\|_1 - \|\widehat{\Omega}_{T_j,j}\|_1 + \|(A_1)_{T_j^c,j}\|_1 + \|\widehat{\Omega}_{T_j,j} - (A_1)_{T_j,j}\|_1 \\
&\le \|(A_1)_{*,j}\|_1 - \|\widehat{\Omega}_{T_j,j}\|_1 + \|(A_1)_{T_j^c,j}\|_1 + \|\widehat{\Omega}_{T_j,j} - (A_1)_{T_j,j}\|_1 \\
&\le 2\|(A_1)_{T_j^c,j}\|_1 + 2\|\widehat{\Omega}_{T_j,j} - (A_1)_{T_j,j}\|_1 \\
&\le 2\|(A_1)_{T_j^c,j}\|_1 + 4\lambda_0\|\Sigma^{-1}\|_1|T_j| \\
&\le (2\lambda_1 + 4\lambda_0\|\Sigma^{-1}\|_1)s_1.
\end{aligned}$$

Suppose $\max_j\sum_{i=1}^{d}|(A_1)_{ij}|^q \le s$ and setting $\lambda_1 = 2\lambda_0\|\Sigma^{-1}\|_1$, we have

$$\lambda_1 s_1 = \max_{1\le j\le d}\sum_{i=1}^{d}\min\{|(A_1)_{ij}|, \lambda_1\} \le \lambda_1\max_{1\le j\le d}\sum_{i=1}^{d}\min\left\{|(A_1)_{ij}|^q/\lambda_1^q, 1\right\} \le \lambda_1^{1-q}s.$$

Therefore, we have

$$\|\widehat{\Omega}_{*,j} - (A_1)_{*,j}\|_1 \leq 4\lambda_1 s_1 \leq 4\lambda_1^{1-q} s = 4s(2\lambda_0\|\Sigma^{-1}\|_1)^{1-q}.$$

Noting that when the lag of the time series $p = 1$, by definition in (12), we have $\widehat{\Omega} = \widehat{A}_1$. This completes the proof. ■

A.2 Proof of the Rest Results

Proof [**Proof of Corollary 1**] Corollary 1 directly follows from Theorem 1, so its proofs is omitted. ■

Proof [**Proof of Corollary 2**] Using the generating model described in Equation (2), we have

$$\begin{aligned}\|X_{T+1} - \widehat{A}_1^{\mathrm{T}} X_T\|_\infty =& \|(A_1^{\mathrm{T}} - \widehat{A}_1^{\mathrm{T}})X_T + Z_{T+1}\|_\infty \\ \leq& \|A_1^{\mathrm{T}} - \widehat{A}_1^{\mathrm{T}}\|_\infty \|X_T\|_\infty + \|Z_{T+1}\|_\infty \\ =& \|A_1 - \widehat{A}_1\|_1 \|X_T\|_\infty + \|Z_{T+1}\|_\infty\end{aligned}$$

Using Lemma 4 in Appendix B, we have

$$\mathbb{P}(\|X_T\|_\infty \leq (\Sigma_{\max}\cdot\alpha\log d)^{1/2}, \|Z_{T+1}\|_\infty \leq (\Psi_{\max}\cdot\alpha\log d)^{1/2}) \geq 1-2(d^{\alpha/2-1}\sqrt{\pi/2\cdot\alpha\log d}\})^{-1}.$$

This, combined with Theorem 1, gives Equation (20). ■

Proof [**Proof of Corollary 3**] Similar as the proof in Corollary 2, we have

$$\begin{aligned}\|X_{T+1} - \bar{A}_1^{\mathrm{T}} X_T\|_2 =& \|(A_1^{\mathrm{T}} - \bar{A}_1^{\mathrm{T}})X_T + Z_{T+1}\|_2 \\ \leq& \|A_1 - \bar{A}_1\|_2 \|X_T\|_2 + \|Z_{T+1}\|_2.\end{aligned}$$

For any Gaussian random vector $Y \sim N_d(0, Q)$, we have $Y = \sqrt{Q}Y_0$ where $Y_0 \sim N_d(0, I_d)$. Using the concentration inequality for Lipschitz functions of standard Gaussian random vector (see, for example, Theorem 3.4 in Massart, 2007), we have

$$\begin{aligned}\mathbb{P}(|\|Y\|_2 - E\|Y\|_2| \geq t) =& \mathbb{P}(|\|\sqrt{Q}Y_0\|_2 - E\|\sqrt{Q}Y_0\|_2| \geq t) \\ \leq& 2\exp\left(-\frac{t^2}{2\|Q\|_2}\right). \end{aligned} \tag{29}$$

Here the inequality exploits the fact that for any vectors $x, y \in \mathbb{R}^d$,

$$|\|\sqrt{Q}x\|_2 - \|\sqrt{Q}y\|_2| \leq \|\sqrt{Q}(x-y)\|_2 \leq \|\sqrt{Q}\|_2\|x-y\|_2,$$

and accordingly the function $x \to \|\sqrt{Q}x\|_2$ has the Lipschitz norm no greater than $\sqrt{\|Q\|_2}$. Using Equation (29), we then have

$$\mathbb{P}(\|X_T\|_2 \leq \sqrt{2\|\Sigma\|_2\log d} + E\|X_T\|_2,\ \|Z_{T+1}\|_2 \leq \sqrt{2\|\Psi\|_2\log d} + E\|Z_{T+1}\|_2) \geq 1 - 4d^{-1}.$$

Finally, we have

$$(E\|Y\|_2)^2 \leq E\|Y\|_2^2 = \mathrm{tr}(\mathrm{Q}).$$

Combined with Theorem 1 and the fact that $\|A_1 - \bar{A}_1\|_2 \leq \|A_1 - \bar{A}_1\|_1$, we have the desired result. ■

Appendix B. Supporting Lemmas

Lemma 3 (Negahban and Wainwright, 2011) *Suppose that $Y \sim N_T(0, Q)$ is a Gaussian random vector. We have, for $\eta > 2T^{-1/2}$,*

$$\mathbb{P}\Big\{\big|\|Y\|_2^2 - E(\|Y\|_2^2)\big| > 4T\eta\|Q\|_2\Big\} \leq 2\exp\Big\{-T(\eta - 2T^{-1/2})^2/2\Big\} + 2\exp(-T/2).$$

Proof [**Proof**] This can be proved by first using the concentration inequality for the Lipschitz functions $\|Y\|_2$ of Gaussian random variables Y. Then combining with the result

$$\|Y\|_2^2 - E(\|Y\|_2^2) \leq (\|Y\|_2 - E\|Y\|_2) \cdot (\|Y\|_2 + E\|Y\|_2),$$

we have the desired concentration inequality. ■

Lemma 4 *Suppose that $Z = (Z_1, \ldots, Z_d)^{\mathrm{T}} \in N_d(0, Q)$ is a Gaussian random vector. Letting $Q_{\max} := \max_i(Q_{ii})$, we have*

$$\mathbb{P}\{\|Z\|_\infty > (Q_{\max} \cdot \alpha \log d)^{1/2}\} \leq \Big(d^{\alpha/2-1}\sqrt{\pi/2 \cdot \alpha \log d}\Big)^{-1}.$$

Proof [**Proof**] Simply using the Gaussian tail probability, we have

$$\mathbb{P}(\|Z\|_\infty > t) \leq \sum_{i=1}^{d} \mathbb{P}(|Z_i| \cdot Q_{ii}^{-1/2} > t \cdot Q_{ii}^{-1/2}) \leq \sum_{i=1}^{d} \frac{2\exp(-t^2/2Q_{ii})}{t \cdot Q_{ii}^{-1/2} \cdot \sqrt{2\pi}} \leq \frac{2d\exp(-t^2/2Q_{\max})}{t \cdot Q_{\max}^{-1/2} \cdot \sqrt{2\pi}}.$$

Taking $t = (Q_{\max} \cdot \alpha \log d)^{1/2}$ into the upper equation, we have the desired result. ■

References

J. H. Ahlberg and E. N. Nilson. Convergence properties of the spline fit. *Journal of the Society for Industrial and Applied Mathematics*, 11(1):95–104, 1963.

J. Bento, M. Ibrahimi, and A. Montanari. Learning networks of stochastic differential equations. In *Advances in Neural Information Processing Systems (NIPS)*, pages 172–180, 2010.

P. J. Bickel and E. Levina. Covariance regularization by thresholding. *The Annals of Statistics*, 36(6):2577–2604, 2008a.

P. J. Bickel and E. Levina. Regularized estimation of large covariance matrices. *The Annals of Statistics*, 36(1):199–227, 2008b.

P. J. Bickel, Y. Ritov, and A. B. Tsybakov. Simultaneous analysis of lasso and Dantzig selector. *The Annals of Statistics*, 37(4):1705–1732, 2009.

R. C. Bradley. Basic properties of strong mixing conditions. A survey and some open questions. *Probability Surveys*, 2:107–144, 2005.

L. Breiman and J. H. Friedman. Predicting multivariate responses in multiple linear regression. *Journal of the Royal Statistical Society: Series B*, 59(1):3–54, 1997.

T. Cai, W. Liu, and X. Luo. A constrained ℓ_1 minimization approach to sparse precision matrix estimation. *Journal of the American Statistical Association*, 106(494):594–607, 2011.

T. T. Cai, C. Zhang, and H. H. Zhou. Optimal rates of convergence for covariance matrix estimation. *The Annals of Statistics*, 38(4):2118–2144, 2010.

E. Candes and T. Tao. The Dantzig selector: statistical estimation when p is much larger than n. *The Annals of Statistics*, 35(6):2313–2351, 2007.

J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010. URL `http://www.jstatsoft.org/v33/i01/`.

C. Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, 37(3):424–438, 1969.

J. D. Hamilton. *Time Series Analysis*, volume 2. Cambridge University Press, 1994.

F. Han and H. Liu. Transition matrix estimation in high dimensional vector autoregressive models. In *International Conference on Machine Learning (ICML)*, pages 172–180, 2013.

S. Haufe, G. Nolte, K. R. Mueller, and N. Krämer. Sparse causal discovery in multivariate time series. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1–16, 2008.

R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 1990.

N. J. Hsu, H. L. Hung, and Y. M. Chang. Subset selection for vector autoregressive processes using lasso. *Computational Statistics and Data Analysis*, 52(7):3645–3657, 2008.

A. B. Kock and L. Callot. Oracle inequalities for high dimensional vector autoregressions. *Journal of Econometrics*, 186(2):325–344, 2015.

Anders Bredahl Kock. On the oracle property of the adaptive lasso in stationary and nonstationary autoregressions. *CREATES Research Papers*, 5, 2012.

X. Li, T. Zhao, X. Yuan, and H. Liu. The flare package for high dimensional linear regression and precision matrix estimation in R. *The Journal of Machine Learning Research*, 16: 553–557, 2015.

P. Loh and M. J. Wainwright. High-dimensional regression with noisy and missing data: Provable guarantees with nonconvexity. *The Annals of Statistics*, 40(3):1637–1664, 2012.

H. Lütkepohl. *New Introduction to Multiple Time Series Analysis*. Cambridge University Press, 2005.

P. Massart. *Concentration Inequalities and Model Selection*. Springer Verlag, 2007.

N. Meinshausen and P. Bühlmann. High-dimensional graphs and variable selection with the lasso. *The Annals of Statistics*, 34(3):1436–1462, 2006.

K. G. Murty. *Linear Programming*. Wiley New York, 1983.

Y. Nardi and A. Rinaldo. Autoregressive process modeling via the lasso procedure. *Journal of Multivariate Analysis*, 102(3):528–549, 2011.

S. Negahban and M. J. Wainwright. Estimation of (near) low-rank matrices with noise and high-dimensional scaling. *The Annals of Statistics*, 39(2):1069–1097, 2011.

H. Qiu, F. Han, H. Liu, and B. Caffo. Joint estimation of multiple graphical models from high dimensional time series. *Journal of the Royal Statistical Society: Series B*, forthcoming.

G. Raskutti, M. J. J Wainwright, and B. Yu. Minimax rates of estimation for high-dimensional linear regression over ℓ_q balls. *IEEE Transactions on Information Theory*, 57(10):6976–6994, 2011.

A. Shojaie and G. Michailidis. Discovering graphical Granger causality using the truncating lasso penalty. *Bioinformatics*, 26(18):i517–i523, 2010.

C. A. Sims. Macroeconomics and reality. *Econometrica*, 48(1):1–48, 1980.

S. Song and P. J. Bickel. Large vector auto regressions. *arXiv preprint arXiv:1106.3915*, 2011.

S. Song, W. K. Härdle, and Y. Ritov. Generalized dynamic semi-parametric factor models for high-dimensional non-stationary time series. *The Econometrics Journal*, 17(2):S101–S131, 2014.

R. S. Tsay. *Analysis of Financial Time Series*. Wiley-Interscience, 2005.

P. A. Valdés-Sosa, J. M. Sánchez-Bornot, A. Lage-Castellanos, M. Vega-Hernández, J. Bosch-Bayard, L. Melie-Garcia, and E. Canales-Rodriguez. Estimating brain functional connectivity with sparse multivariate autoregression. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 360(1457):969–981, 2005.

J. M. Varah. A lower bound for the smallest singular value of a matrix. *Linear Algebra and Its Applications*, 11(1):3–5, 1975.

V. Q. Vu and J. Lei. Minimax rates of estimation for sparse PCA in high dimensions. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1278–1286, 2012.

M. J. Wainwright. Sharp thresholds for noisy and high-dimensional recovery of sparsity using ℓ_1-constrained quadratic programming (lasso). *IEEE Transactions on Information Theory*, 55(5):2183–2202, 2009.

H. Wang, G. Li, and C. L. Tsai. Regression coefficient and autoregressive order shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B*, 69(1):63–78, 2007.

I. Weiner, N. Schmitt, and S. Highhouse. *Handbook of Psychology, Industrial and Organizational Psychology*. John Wiley and Sons, 2012.

M. Yuan. High dimensional inverse covariance matrix estimation via linear programming. *The Journal of Machine Learning Research*, 11:2261–2286, 2010.

P. Zhao and B. Yu. On model selection consistency of lasso. *The Journal of Machine Learning Research*, 7:2541–2563, 2006.

H. Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429, 2006.

Journal of Machine Learning Research 16 (2015) 3151-3181 Submitted 9/14; Revised 7/15; Published 12/15

Global Convergence of Online Limited Memory BFGS

Aryan Mokhtari ARYANM@SEAS.UPENN.EDU
Alejandro Ribeiro ARIBEIRO@SEAS.UPENN.EDU
Department of Electrical and Systems Engineering
University of Pennsylvania
Philadelphia, PA 19104, USA

Editor: Léon Bottou

Abstract

Global convergence of an online (stochastic) limited memory version of the Broyden-Fletcher-Goldfarb-Shanno (BFGS) quasi-Newton method for solving optimization problems with stochastic objectives that arise in large scale machine learning is established. Lower and upper bounds on the Hessian eigenvalues of the sample functions are shown to suffice to guarantee that the curvature approximation matrices have bounded determinants and traces, which, in turn, permits establishing convergence to optimal arguments with probability 1. Experimental evaluation on a search engine advertising problem showcase reductions in convergence time relative to stochastic gradient descent algorithms.

Keywords: quasi-Newton methods, large-scale optimization, stochastic optimization

1. Introduction

Many problems in Machine Learning can be reduced to the minimization of a stochastic objective defined as an expectation over a set of random functions (Bottou and Le Cun (2005); Bottou (2010); Shalev-Shwartz and Srebro (2008); Mokhtari and Ribeiro (2014b)). Specifically, consider an optimization variable $\mathbf{w} \in \mathbb{R}^n$ and a random variable $\boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}^p$ that determines the choice of a function $f(\mathbf{w}, \boldsymbol{\theta}) : \mathbb{R}^{n \times p} \to \mathbb{R}$. Stochastic optimization problems entail determination of the argument $\mathbf{w}^*$ that minimizes the expected value $F(\mathbf{w}) := \mathbb{E}_{\boldsymbol{\theta}}[f(\mathbf{w}, \boldsymbol{\theta})]$,

$$\mathbf{w}^* \;:=\; \operatorname*{argmin}_{\mathbf{w}} \mathbb{E}_{\boldsymbol{\theta}}[f(\mathbf{w}, \boldsymbol{\theta})] \;:=\; \operatorname*{argmin}_{\mathbf{w}} F(\mathbf{w}). \tag{1}$$

We refer to $f(\mathbf{w}, \boldsymbol{\theta})$ as the random or instantaneous functions and to $F(\mathbf{w}) := \mathbb{E}_{\boldsymbol{\theta}}[f(\mathbf{w}, \boldsymbol{\theta})]$ as the average function. A canonical class of problems having this form are support vector machines (SVMs) that reduce binary classification to the determination of a hyperplane that separates points in a given training set; see, e.g., (Vapnik (2000); Bottou (2010); Boser et al. (1992)). In that case, $\boldsymbol{\theta}$ denotes individual training samples, $f(\mathbf{w}, \boldsymbol{\theta})$ the loss of choosing the hyperplane defined by $\mathbf{w}$, and $F(\mathbf{w}) := \mathbb{E}_{\boldsymbol{\theta}}[f(\mathbf{w}, \boldsymbol{\theta})]$ the mean loss across all elements of the training set. The optimal argument $\mathbf{w}^*$ is the optimal linear classifier.

Numerical evaluation of objective function gradients $\nabla_{\mathbf{w}} F(\mathbf{w}) = \mathbb{E}_{\boldsymbol{\theta}}[\nabla_{\mathbf{w}} f(\mathbf{w}, \boldsymbol{\theta})]$ is intractable when the cardinality of Θ is large, as is the case, e.g., when SVMs are trained on large sets. This motivates the use of algorithms relying on stochastic gradients that provide gradient estimates based on small data subsamples. For the purpose of this paper stochastic optimization algorithms can be divided into three categories: Stochastic gradient descent (SGD) and related first order methods, stochastic Newton methods, and stochastic quasi-Newton methods.

SGD is the most popular method used to solve stochastic optimization problems (Bottou (2010); Shalev-Shwartz et al. (2011); Zhang (2004)). However, as we consider problems of ever larger dimension their slow convergence times have limited their practical appeal and fostered the search for alternatives. In this regard, it has to be noted that SGD is slow because of the use of gradients

as descent directions which leads to poor curvature approximation in ill-conditioned problems. The golden standard to deal with these ill-conditioned functions in a deterministic setting is Newton's method. However, unbiased stochastic estimates of Newton steps can't be computed in general. This fact limits the application of stochastic Newton methods to problems with specific structure (Birge et al. (1994); Zargham et al. (2013)).

If SGD is slow to converge and stochastic Newton can't be used in general, the remaining alternative is to modify deterministic quasi-Newton methods that speed up convergence times relative to gradient descent without using Hessian evaluations (Dennis and Moré (1974); Powell (1976); Byrd et al. (1987); Nocedal and Wright (1999)). This has resulted in the development of the stochastic quasi-Newton methods known as online (o) Broyden-Fletcher-Goldfarb-Shanno (BFGS) (Schraudolph et al. (2007); Bordes et al. (2009)), regularized stochastic BFGS (RES) (Mokhtari and Ribeiro (2014a)), and online limited memory (oL)BFGS (Schraudolph et al. (2007)) which occupy the middle ground of broad applicability irrespective of problem structure and conditioning. All three of these algorithms extend BFGS by using stochastic gradients both as descent directions and constituents of Hessian estimates. The oBFGS algorithm is a direct generalization of BFGS that uses stochastic gradients in lieu of deterministic gradients. RES differs in that it further modifies BFGS to yield an algorithm that retains its convergence advantages while improving theoretical convergence guarantees and numerical behavior. The oLBFGS method uses a modification of BFGS to reduce the computational cost of each iteration.

An important observation here is that in trying to adapt to the changing curvature of the objective, stochastic quasi-Newton methods may end up exacerbating the problem. Indeed, since Hessian estimates are stochastic, it is possible to end up with almost singular Hessian estimates. The corresponding small eigenvalues then result in a catastrophic amplification of the noise which nullifies progress made towards convergence. This is not a minor problem. In oBFGS this possibility precludes convergence analyses and may result in erratic numerical behavior (Mokhtari and Ribeiro (2014a)). As a matter of fact, the main motivation for the introduction of RES is to avoid this catastrophic noise amplification so as to retain smaller convergence times while ensuring that optimal arguments are found with probability 1 (Mokhtari and Ribeiro (2014a)). Generally, stochastic quasi-Newton methods whose Hessian approximations have bounded eigenvalues converge to optimal arguments (Sunehag et al. (2009)). However valuable, the convergence guarantees of RES and the convergence time advantages of oBFGS and RES are tainted by an iteration cost of order $O(n^2)$ and $O(n^3)$, respectively, which precludes their use in problems where n is very large. In deterministic settings this problem is addressed by limited memory (L)BFGS (Liu and Nocedal (1989)) which can be easily generalized to develop the oLBFGS algorithm (Schraudolph et al. (2007)). Numerical tests of oLBFGS are promising but theoretical convergence characterizations are still lacking. The main contribution of this paper is to show that oLBFGS converges with probability 1 to optimal arguments across realizations of the random variables θ. This is the same convergence guarantee provided for RES and is in marked contrast with oBFGS, which fails to converge if not properly regularized. Convergence guarantees for oLBFGS do not require such measures.

We begin the paper with brief discussions of deterministic BFGS (Section 2) and LBFGS (Section 2.1) and the introduction of oLBFGS (Section 2.2). The fundamental idea in BFGS and oLBFGS is to continuously satisfy a secant condition while staying close to previous curvature estimates. They differ in that BFGS uses all past gradients to estimate curvature while oLBFGS uses a fixed moving window of past gradients. The use of this window reduces memory and computational cost (Appendix A). The difference between LBFGS and oLBFGS is the use of stochastic gradients in lieu of their deterministic counterparts. Note that choosing large mini-batch for computing stochastic gradients reduces the variance of stochastic approximation and decreases the gap between LBFGS and oLBFGS, however, increases the computational cost of oLBFGS. Therefore, picking a suitable mini-batch size is an important step in the implementation of oLBFGS.

Convergence properties of oLBFGS are then analyzed (Section 3). Under the assumption that the sample functions $f(\mathbf{w}, \boldsymbol{\theta})$ are strongly convex we show that the trace and determinant of the

Hessian approximations computed by oLBFGS are upper and lower bounded, respectively (Lemma 3). These bounds are then used to limit the range of variation of the ratio between the Hessian approximations' largest and smallest eigenvalues (Lemma 4). In turn, this condition number limit is shown to be sufficient to prove convergence to the optimal argument $\mathbf{w}^*$ with probability 1 over realizations of the sample functions (Theorem 6). This is an important result because it ensures that oLBFGS doesn't suffer from the numerical problems that hinder oBFGS. We complement this almost sure convergence result with a characterization of the convergence rate which is shown to be at least $O(1/t)$ in expectation (Theorem 7). It is fair to emphasize that, different from the deterministic case, the convergence rate of oLBFGS is not better than the convergence rate of SGD. This is not a limitation of our analysis. The difference between stochastic and regular gradients introduces a noise term that dominates convergence once we are close to the optimum, which is where superlinear convergence rates manifest. In fact, the same convergence rate would be observed if exact Hessians were available. The best that can be proven of oLBFGS is that the convergence rate is not worse than that of SGD. Given that theoretical guarantees only state that the curvature correction does not exacerbate the problem's condition it is perhaps fairer to describe oLBFGS as an adaptive reconditioning strategy instead of a stochastic quasi-Newton method. The latter description refers to the genesis of the algorithm. The former is a more accurate description of its actual behavior.

To show the advantage of oLBFGS we use it to train a logistic regressor to predict the click through rate in a search engine advertising problem (Section 4). The logistic regression uses a heterogeneous feature vector with 174,026 binary entries that describe the user, the search, and the advertisement (Section 4.1). Being a large scale problem with heterogeneous data, the condition number of the logistic log likelihood objective is large and we expect to see significant advantages of oLBFGS relative to SGD. This expectation is fulfilled. The oLBFGS algorithm trains the regressor using less than 1% of the data required by SGD to obtain similar classification accuracy. (Section 4.3). We close the paper with concluding remarks (Section 5).

Notation Lowercase boldface $\mathbf{v}$ denotes a vector and uppercase boldface $\mathbf{A}$ a matrix. We use $\|\mathbf{v}\|$ to denote the Euclidean norm of vector $\mathbf{v}$ and $\|\mathbf{A}\|$ to denote the Euclidean norm of matrix $\mathbf{A}$. The trace of $\mathbf{A}$ is written as $\mathrm{tr}(\mathbf{A})$ and the determinant as $\det(\mathbf{A})$. We use $\mathbf{I}$ for the identity matrix of appropriate dimension. The notation $\mathbf{A} \succeq \mathbf{B}$ implies that the matrix $\mathbf{A} - \mathbf{B}$ is positive semidefinite. The operator $\mathbb{E}_{\mathbf{x}}[\cdot]$ stands in for expectation over random variable $\mathbf{x}$ and $\mathbb{E}[\cdot]$ for expectation with respect to the distribution of a stochastic process.

2. Algorithm Definition

Recall the definitions of the sample functions $f(\mathbf{w}, \boldsymbol{\theta})$ and the average function $F(\mathbf{w}) := \mathbb{E}_{\boldsymbol{\theta}}[f(\mathbf{w}, \boldsymbol{\theta})]$. We assume the sample functions $f(\mathbf{w}, \boldsymbol{\theta})$ are strongly convex for all $\boldsymbol{\theta}$. This implies the objective function $F(\mathbf{w}) := \mathbb{E}_{\boldsymbol{\theta}}[f(\mathbf{w}, \boldsymbol{\theta})]$, being an average of the strongly convex sample functions, is also strongly convex. We define the gradient $\mathbf{s}(\mathbf{w}) := \nabla F(\mathbf{w})$ of the average function $F(\mathbf{w})$ and assume that it can be computed as

$$\mathbf{s}(\mathbf{w}) := \nabla F(\mathbf{w}) = \mathbb{E}_{\boldsymbol{\theta}}[\nabla f(\mathbf{w}, \boldsymbol{\theta})]. \tag{2}$$

Since the function $F(\mathbf{w})$ is strongly convex, gradients $\mathbf{s}(\mathbf{w})$ are descent directions that can be used to find the optimal argument $\mathbf{w}^*$ in (1). Introduce then a time index t, a step size ϵ_t, and a positive definite matrix $\mathbf{B}_t^{-1} \succ 0$ to define a generic descent algorithm through the iteration

$$\mathbf{w}_{t+1} \;=\; \mathbf{w}_t - \epsilon_t \, \mathbf{B}_t^{-1} \mathbf{s}(\mathbf{w}_t) \;=\; \mathbf{w}_t - \epsilon_t \, \mathbf{d}_t. \tag{3}$$

where we have also defined the descent step $\mathbf{d}_t = \mathbf{B}_t^{-1} \mathbf{s}(\mathbf{w}_t)$. When $\mathbf{B}_t^{-1} = \mathbf{I}$ is the identity matrix, (3) reduces to gradient descent. When $\mathbf{B}_t = \mathbf{H}(\mathbf{w}_t) := \nabla^2 F(\mathbf{w}_t)$ is the Hessian of the objective function, (3) defines Newton's algorithm. In this paper we focus on quasi-Newton methods whereby we attempt to select matrices $\mathbf{B}_t$ close to the Hessian $\mathbf{H}(\mathbf{w}_t)$. Various methods are known to select

matrices $\mathbf{B}_t$, including those by Broyden e.g., Broyden et al. (1973); Davidon, Fletcher, and Powell (DFP) e.g., Fletcher (2013); and Broyden, Fletcher, Goldfarb, and Shanno (BFGS) e.g., Byrd et al. (1987); Powell (1976). We work with the matrices $\mathbf{B}_t$ used in BFGS since they have been observed to work best in practice (see Byrd et al. (1987)).

In BFGS, the function's curvature $\mathbf{B}_t$ is approximated by a finite difference. Let $\mathbf{v}_t$ denote the variable variation at time t and $\mathbf{r}_t$ the gradient variation at time t which are respectively defined as

$$\mathbf{v}_t := \mathbf{w}_{t+1} - \mathbf{w}_t, \qquad \mathbf{r}_t := \mathbf{s}(\mathbf{w}_{t+1}) - \mathbf{s}(\mathbf{w}_t). \tag{4}$$

We select the matrix $\mathbf{B}_{t+1}$ to be used in the next time step so that it satisfies the secant condition $\mathbf{B}_{t+1}\mathbf{v}_t = \mathbf{r}_t$. The rationale for this selection is that the Hessian $\mathbf{H}(\mathbf{w}_t)$ satisfies this condition for $\mathbf{w}_{t+1}$ tending to $\mathbf{w}_t$. Notice however that the secant condition $\mathbf{B}_{t+1}\mathbf{v}_t = \mathbf{r}_t$ is not enough to completely specify $\mathbf{B}_{t+1}$. To resolve this indeterminacy, matrices $\mathbf{B}_{t+1}$ in BFGS are also required to be as close as possible to the previous Hessian approximation $\mathbf{B}_t$ in terms of differential entropy (see Mokhtari and Ribeiro (2014a)). These conditions can be resolved in closed form leading to the explicit expression,

$$\mathbf{B}_{t+1} = \mathbf{B}_t + \frac{\mathbf{r}_t\mathbf{r}_t^T}{\mathbf{v}_t^T\mathbf{r}_t} - \frac{\mathbf{B}_t\mathbf{v}_t\mathbf{v}_t^T\mathbf{B}_t}{\mathbf{v}_t^T\mathbf{B}_t\mathbf{v}_t}. \tag{5}$$

While the expression in (5) permits updating the Hessian approximations $\mathbf{B}_{t+1}$, implementation of the descent step in (3) requires its inversion. This can be avoided by using the Sherman-Morrison formula in (5) to write

$$\mathbf{B}_{t+1}^{-1} \;=\; \mathbf{Z}_t^T\;\mathbf{B}_t^{-1}\;\mathbf{Z}_t \;+\; \rho_t\;\mathbf{v}_t\mathbf{v}_t^T, \tag{6}$$

where we defined the scalar ρ_t and the matrix $\mathbf{Z}_t$ as

$$\rho_t \;:=\; \frac{1}{\mathbf{v}_t^T\mathbf{r}_t}\;, \qquad \mathbf{Z}_t \;:=\; \mathbf{I} - \rho_t\mathbf{r}_t\mathbf{v}_t^T. \tag{7}$$

The updates in (5) and (6) require the inner product of the gradient and variable variations to be positive, i.e., $\mathbf{v}_t^T\mathbf{r}_t > 0$. This is always true if the objective $F(\mathbf{w})$ is strongly convex and further implies that $\mathbf{B}_{t+1}^{-1}$ stays positive definite if $\mathbf{B}_t^{-1} \succ \mathbf{0}$, (Nocedal and Wright (1999)).

Each BFGS iteration has a cost of $O(n^2)$ arithmetic operations. This is less than the $O(n^3)$ of each step in Newton's method but more than the $O(n)$ cost of each gradient descent iteration. In general, the relative convergence rates are such that the total computational cost of BFGS to achieve a target accuracy is smaller than the corresponding cost of gradient descent. Still, alternatives to reduce the computational cost of each iteration are of interest for large scale problems. Likewise, BFGS requires storage and propagation of the $O(n^2)$ elements of $\mathbf{B}_t^{-1}$, whereas gradient descent requires storage of $O(n)$ gradient elements only. This motivates alternatives that have smaller memory footprints. Both of these objectives are accomplished by the limited memory (L)BFGS algorithm that we describe in the following section.

2.1 LBFGS: Limited Memory BFGS

As it follows from (6), the updated Hessian inverse approximation $\mathbf{B}_t^{-1}$ depends on $\mathbf{B}_{t-1}^{-1}$ and the curvature information pairs $\{\mathbf{v}_{t-1},\, \mathbf{r}_{t-1}\}$. In turn, to compute $\mathbf{B}_{t-1}^{-1}$, the estimate $\mathbf{B}_{t-2}^{-1}$ and the curvature pair $\{\mathbf{v}_{t-2}, \mathbf{r}_{t-2}\}$ are used. Proceeding recursively, it follows that $\mathbf{B}_t^{-1}$ is a function of the initial approximation $\mathbf{B}_0^{-1}$ and all previous t curvature information pairs $\{\mathbf{v}_u,\, \mathbf{r}_u\}_{u=0}^{t-1}$. The idea in LBFGS is to restrict the use of past curvature information to the last τ pairs $\{\mathbf{v}_u, \mathbf{r}_u\}_{u=t-\tau}^{t-1}$. Since earlier iterates $\{\mathbf{v}_u, \mathbf{r}_u\}$ with $u < t - \tau$ are likely to carry little information about the curvature at the current iterate $\mathbf{w}_t$, this restriction is expected to result in a minimal performance penalty.

For a precise definition, pick a positive definite matrix $\mathbf{B}_{t,0}^{-1}$ as the initial Hessian inverse approximation at step t. Proceed then to perform τ updates of the form in (6) using the last τ curvature

information pairs $\{\mathbf{v}_u, \mathbf{r}_u\}_{u=t-\tau}^{t-1}$. Denoting as $\mathbf{B}_{t,u}^{-1}$ the curvature approximation after u updates are performed we have that the refined matrix approximation $\mathbf{B}_{t,u+1}^{-1}$ is given by [cf. (6)]

$$\mathbf{B}_{t,u+1}^{-1} \;=\; \mathbf{Z}_{t-\tau+u}^T\, \mathbf{B}_{t,u}^{-1}\, \mathbf{Z}_{t-\tau+u} \;+\; \rho_{t-\tau+u}\, \mathbf{v}_{t-\tau+u}\, \mathbf{v}_{t-\tau+u}^T, \tag{8}$$

where $u = 0, \ldots, \tau - 1$ and the constants $\rho_{t-\tau+u}$ and rank-one plus identity matrices $\mathbf{Z}_{t-\tau+u}$ are as given in (7). The inverse Hessian approximation $\mathbf{B}_t^{-1}$ to be used in (3) is the one yielded after completing the τ updates in (8), i.e., $\mathbf{B}_t^{-1} = \mathbf{B}_{t,\tau}^{-1}$. Observe that when $t < \tau$ there are not enough pairs $\{\mathbf{v}_u, \mathbf{r}_u\}$ to perform τ updates. In such case we just redefine $\tau = t$ and proceed to use the $t = \tau$ available pairs $\{\mathbf{v}_u, \mathbf{r}_u\}_{u=0}^{t-1}$.

Implementation of the product $\mathbf{B}_t^{-1}\mathbf{s}(\mathbf{w}_t)$ in (3) for matrices $\mathbf{B}_t^{-1} = \mathbf{B}_{t,\tau}^{-1}$ obtained from the recursion in (8) does not need explicit computation of the matrix $\mathbf{B}_{t,\tau}^{-1}$. Although the details are not straightforward, observe that each iteration in (8) is similar to a rank-one update and that as such it is not unreasonable to expect that the product $\mathbf{B}_t^{-1}\mathbf{s}(\mathbf{w}_t) = \mathbf{B}_{t,\tau}^{-1}\mathbf{s}(\mathbf{w}_t)$ can be computed using τ recursive inner products. Assuming that this is possible, the implementation of the recursion in (8) doesn't need computation and storage of prior matrices $\mathbf{B}_{t-1}^{-1}$. Rather, it suffices to keep the τ most recent curvature information pairs $\{\mathbf{v}_u, \mathbf{r}_u\}_{u=t-\tau}^{t-1}$, thus reducing storage requirements from $O(n^2)$ to $O(\tau n)$. Furthermore, each of these inner products can be computed at a cost of n operations yielding a total computational cost of $O(\tau n)$ per LBFGS iteration. Hence, LBFGS decreases both the memory requirements and the computational cost of each iteration from the $O(n^2)$ required by regular BFGS to $O(\tau n)$. We present the details of this iteration in the context of the online (stochastic) LBFGS that we introduce in the following section.

2.2 Online (Stochastic) Limited Memory BFGS

To implement (3) and (8) we need to compute gradients $\mathbf{s}(\mathbf{w}_t)$. This is impractical when the number of functions $f(\mathbf{w}, \boldsymbol{\theta})$ is large, as is the case in most stochastic problems of practical interest and motivates the use of stochastic gradients in lieu of actual gradients. Consider a given set of L realizations $\tilde{\boldsymbol{\theta}} = [\boldsymbol{\theta}_1; ...; \boldsymbol{\theta}_L]$ and define the stochastic gradient of $F(\mathbf{w})$ at $\mathbf{w}$ given samples $\tilde{\boldsymbol{\theta}}$ as

$$\hat{\mathbf{s}}(\mathbf{w}, \tilde{\boldsymbol{\theta}}) := \frac{1}{L} \sum_{l=1}^{L} \nabla f(\mathbf{w}, \boldsymbol{\theta}_l). \tag{9}$$

In oLBFGS we use stochastic gradients $\hat{\mathbf{s}}(\mathbf{w}, \tilde{\boldsymbol{\theta}})$ for descent directions and curvature estimators. In particular, the descent iteration in (3) is replaced by the descent iteration

$$\mathbf{w}_{t+1} \;=\; \mathbf{w}_t - \epsilon_t\, \hat{\mathbf{B}}_t^{-1} \hat{\mathbf{s}}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t) \;=\; \mathbf{w}_t - \epsilon_t \hat{\mathbf{d}}_t, \tag{10}$$

where $\tilde{\boldsymbol{\theta}}_t = [\boldsymbol{\theta}_{t1}; ...; \boldsymbol{\theta}_{tL}]$ is the set of samples used at step t to compute the stochastic gradient $\hat{\mathbf{s}}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t)$ as per (9) and the matrix $\hat{\mathbf{B}}_t^{-1}$ is a function of past stochastic gradients $\hat{\mathbf{s}}(\mathbf{w}_u, \tilde{\boldsymbol{\theta}}_u)$ with $u \leq t$ instead of a function of past gradients $\mathbf{s}(\mathbf{w}_u)$ as in (3). As we also did in (3) we have defined the stochastic step $\hat{\mathbf{d}}_t := \hat{\mathbf{B}}_t^{-1}\hat{\mathbf{s}}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t)$ to simplify upcoming discussions.

To properly specify $\hat{\mathbf{B}}_t^{-1}$ we define the stochastic gradient variation $\hat{\mathbf{r}}_t$ at time t as the difference between the stochastic gradients $\hat{\mathbf{s}}(\mathbf{w}_{t+1}, \tilde{\boldsymbol{\theta}}_t)$ and $\hat{\mathbf{s}}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t)$ associated with subsequent iterates $\mathbf{w}_{t+1}$ and $\mathbf{w}_t$ and the *common* set of samples $\tilde{\boldsymbol{\theta}}_t$ [cf. (4)],

$$\hat{\mathbf{r}}_t := \hat{\mathbf{s}}(\mathbf{w}_{t+1}, \tilde{\boldsymbol{\theta}}_t) - \hat{\mathbf{s}}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t). \tag{11}$$

Observe that $\hat{\mathbf{s}}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t)$ is the stochastic gradient used at time t in (10) but that $\hat{\mathbf{s}}(\mathbf{w}_{t+1}, \tilde{\boldsymbol{\theta}}_t)$ is computed solely for the purpose of determining the stochastic gradient variation. The perhaps more natural definition $\hat{\mathbf{s}}(\mathbf{w}_{t+1}, \tilde{\boldsymbol{\theta}}_{t+1}) - \hat{\mathbf{s}}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t)$ for the stochastic gradient variation, which relies on the

stochastic gradient $\hat{\mathbf{s}}(\mathbf{w}_{t+1}, \tilde{\boldsymbol{\theta}}_{t+1})$ used at time $t+1$ in (10) is not sufficient to guarantee convergence; see e.g.,(Mokhtari and Ribeiro (2014a)).

To define the oLBFGS algorithm we just need to provide stochastic versions of the definitions in (7) and (8). The scalar constants and identity plus rank-one matrices in (7) are redefined to the corresponding stochastic quantities

$$\hat{\rho}_{t-\tau+u} \;=\; \frac{1}{\mathbf{v}_{t-\tau+u}^T \hat{\mathbf{r}}_{t-\tau+u}} \quad \text{and} \quad \hat{\mathbf{Z}}_{t-\tau+u} \;=\; \mathbf{I} - \hat{\rho}_{t-\tau+u} \hat{\mathbf{r}}_{t-\tau+u} \mathbf{v}_{t-\tau+u}^T, \tag{12}$$

whereas the LBFGS matrix $\mathbf{B}_t^{-1} = \mathbf{B}_{t,\tau}^{-1}$ in (8) is replaced by the oLBFGS Hessian inverse approximation $\hat{\mathbf{B}}_t^{-1} = \hat{\mathbf{B}}_{t,\tau}^{-1}$ which we define as the outcome of τ recursive applications of the update,

$$\hat{\mathbf{B}}_{t,u+1}^{-1} \;=\; \hat{\mathbf{Z}}_{t-\tau+u}^T \, \hat{\mathbf{B}}_{t,u}^{-1} \, \hat{\mathbf{Z}}_{t-\tau+u} + \hat{\rho}_{t-\tau+u} \, \mathbf{v}_{t-\tau+u} \, \mathbf{v}_{t-\tau+u}^T, \tag{13}$$

where the initial matrix $\hat{\mathbf{B}}_{t,0}^{-1}$ is given and the time index is $u = 0, \ldots, \tau-1$. The oLBFGS algorithm is defined by the stochastic descent iteration in (10) with matrices $\hat{\mathbf{B}}_t^{-1} = \hat{\mathbf{B}}_{t,\tau}^{-1}$ computed by τ recursive applications of (13). Except for the fact that they use stochastic variables, (10) and (13) are identical to (3) and (8). Thus, as is the case in (3), the Hessian inverse approximation $\hat{\mathbf{B}}_t^{-1}$ in (13) is a function of the initial Hessian inverse approximation $\mathbf{B}_{t,0}^{-1}$ and the τ most recent curvature information pairs $\{\mathbf{v}_u, \hat{\mathbf{r}}_u\}_{u=t-\tau}^{t-1}$. Likewise, when $t < \tau$ there are not enough pairs $\{\mathbf{v}_u, \hat{\mathbf{r}}_u\}$ to perform τ updates. In such case we just redefine $\tau = t$ and proceed to use the $t = \tau$ available pairs $\{\mathbf{v}_u, \hat{\mathbf{r}}_u\}_{u=0}^{t-1}$. We also point out that the update in (13) necessitates $\hat{\mathbf{r}}_u^T \mathbf{v}_u > 0$ for all time indexes u. This is true as long as the instantaneous functions $f(\mathbf{w}, \boldsymbol{\theta})$ are strongly convex with respect to $\mathbf{w}$ as we show in Lemma 2.

The equations in (10) and (13) are used conceptually but not in practical implementations. For the latter we exploit the structure of (13) to rearrange the terms in the computation of the product $\hat{\mathbf{B}}_t^{-1} \hat{\mathbf{s}}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t)$. To see how this is done consider the recursive update for the Hessian inverse approximation $\hat{\mathbf{B}}_t^{-1}$ in (13) and make $u = \tau - 1$ to write

$$\hat{\mathbf{B}}_t^{-1} \;=\; \hat{\mathbf{B}}_{t,\tau}^{-1} \;=\; \left(\hat{\mathbf{Z}}_{t-1}^T\right) \hat{\mathbf{B}}_{t,\tau-1}^{-1} \left(\hat{\mathbf{Z}}_{t-1}\right) + \hat{\rho}_{t-1} \, \mathbf{v}_{t-1} \, \mathbf{v}_{t-1}^T. \tag{14}$$

Equation (14) shows the relation between the Hessian inverse approximation $\hat{\mathbf{B}}_t^{-1}$ and the $(\tau-1)$st updated version of the initial Hessian inverse approximation $\hat{\mathbf{B}}_{t,\tau-1}^{-1}$ at step t. Set now $u = \tau - 2$ in (13) to express $\hat{\mathbf{B}}_{t,\tau-1}^{-1}$ in terms of $\hat{\mathbf{B}}_{t,\tau-2}^{-1}$ and substitute the result in (14) to rewrite $\hat{\mathbf{B}}_t^{-1}$ as

$$\hat{\mathbf{B}}_t^{-1} \;=\; \left(\hat{\mathbf{Z}}_{t-1}^T \hat{\mathbf{Z}}_{t-2}^T\right) \hat{\mathbf{B}}_{t,\tau-2}^{-1} \left(\hat{\mathbf{Z}}_{t-2} \hat{\mathbf{Z}}_{t-1}\right) + \; \hat{\rho}_{t-2} \left(\hat{\mathbf{Z}}_{t-1}^T\right) \mathbf{v}_{t-2} \, \mathbf{v}_{t-2}^T \left(\hat{\mathbf{Z}}_{t-1}\right) + \; \hat{\rho}_{t-1} \, \mathbf{v}_{t-1} \, \mathbf{v}_{t-1}^T. \tag{15}$$

We can proceed recursively by substituting $\hat{\mathbf{B}}_{t,\tau-2}^{-1}$ for its expression in terms of $\hat{\mathbf{B}}_{t,\tau-3}^{-1}$ and in the result substitute $\hat{\mathbf{B}}_{t,\tau-3}^{-1}$ for its expression in terms of $\hat{\mathbf{B}}_{t,\tau-3}^{-1}$ and so on. Observe that a new summand is added in each of these substitutions from which it follows that repeating this process τ times yields

$$\begin{aligned}\hat{\mathbf{B}}_t^{-1} = \; & \left(\hat{\mathbf{Z}}_{t-1}^T \ldots \hat{\mathbf{Z}}_{t-\tau}^T\right) \hat{\mathbf{B}}_{t,0}^{-1} \left(\hat{\mathbf{Z}}_{t-\tau} \ldots \hat{\mathbf{Z}}_{t-1}\right) + \; \hat{\rho}_{t-\tau} \left(\hat{\mathbf{Z}}_{t-1}^T \ldots \hat{\mathbf{Z}}_{t-\tau+1}^T\right) \mathbf{v}_{t-\tau} \mathbf{v}_{t-\tau}^T \left(\hat{\mathbf{Z}}_{t-\tau+1} \ldots \hat{\mathbf{Z}}_{t-1}\right) \\ & + \; \cdots + \; \hat{\rho}_{t-2} \left(\hat{\mathbf{Z}}_{t-1}^T\right) \mathbf{v}_{t-2} \mathbf{v}_{t-2}^T \left(\hat{\mathbf{Z}}_{t-1}\right) + \; \hat{\rho}_{t-1} \mathbf{v}_{t-1} \mathbf{v}_{t-1}^T. \end{aligned} \tag{16}$$

The important observation in (16) is that the matrix $\hat{\mathbf{Z}}_{t-1}$ and its transpose $\hat{\mathbf{Z}}_{t-1}^T$ are the first and last product terms of all summands except the last, that the matrices $\hat{\mathbf{Z}}_{t-2}$ and its transpose $\hat{\mathbf{Z}}_{t-2}^T$ are second and penultimate in all terms but the last two, and so on. Thus, when computing the

oLBFGS step $\hat{\mathbf{d}}_t := \hat{\mathbf{B}}_t^{-1}\hat{\mathbf{s}}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t)$ the operations needed to compute the product with the next to last summand of (16) can be reused to compute the product with the second to last summand which in turn can be reused in determining the product with the third to last summand and so on. This observation compounded with the fact that multiplications with the identity plus rank one matrices $\hat{\mathbf{Z}}_{t-1}$ requires $O(n)$ operations yields an algorithm that can compute the oLBFGS step $\hat{\mathbf{d}}_t := \hat{\mathbf{B}}_t^{-1}\hat{\mathbf{s}}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t)$ in $O(\tau n)$ operations.

We summarize the specifics of such computation in the following proposition where we consider the computation of the product $\hat{\mathbf{B}}_t^{-1}\mathbf{p}$ with a given arbitrary vector $\mathbf{p}$.

Proposition 1 *Consider the oLBFGS Hessian inverse approximation $\hat{\mathbf{B}}_t^{-1} = \hat{\mathbf{B}}_{t,\tau}^{-1}$ obtained after τ recursive applications of the update in* (13) *with the scalar sequence $\hat{\rho}_{t-\tau+u}$ and identity plus rank-one matrix sequence $\hat{\mathbf{Z}}_{t-\tau+u}$ as defined in* (12) *for given variable and stochastic gradient variation pairs $\{\mathbf{v}_u, \mathbf{r}_u\}_{u=t-\tau}^{t-1}$. For a given vector $\mathbf{p} = \mathbf{p}_0$ define the sequence of vectors $\mathbf{p}_k$ through the recursion*

$$\mathbf{p}_{u+1} \;=\; \mathbf{p}_u - \alpha_u \hat{\mathbf{r}}_{t-u-1} \qquad \text{for } u = 0, \ldots, \tau - 1, \tag{17}$$

where we also define the constants $\alpha_u := \hat{\rho}_{t-u-1}\mathbf{v}_{t-u-1}^T\mathbf{p}_u$. Further define the sequence of vectors $\mathbf{q}_k$ with initial value $\mathbf{q}_0 = \hat{\mathbf{B}}_{t,0}^{-1}\mathbf{p}_\tau$ and subsequent elements

$$\mathbf{q}_{u+1} \;=\; \mathbf{q}_u + (\alpha_{\tau-u-1} - \beta_u)\mathbf{v}_{t-\tau+u} \quad \text{for } u = 0, \ldots, \tau - 1, \tag{18}$$

where we define constants $\beta_u := \hat{\rho}_{t-\tau+u}\hat{\mathbf{r}}_{t-\tau+u}^T\mathbf{q}_u$. The product $\hat{\mathbf{B}}_t^{-1}\mathbf{p}$ equals $\mathbf{q}_\tau$, i.e., $\hat{\mathbf{B}}_t^{-1}\mathbf{p} \;=\; \mathbf{q}_\tau$.

Proof See Appendix A. ■

The reorganization of computations described in Proposition 1 has been done for the deterministic LBFGS method in, e.g., (Nocedal and Wright (1999)). We have used the same technique here for computing the descent direction of oLBFGS and have shown the result and derivations for completeness. In any event, Proposition 1 asserts that it is possible to reduce the computation of the product $\hat{\mathbf{B}}_t^{-1}\mathbf{p}$ between the oLBFGS Hessian approximation matrix and arbitrary vector $\mathbf{p}$ to the computation of two vector sequences $\{\mathbf{p}_u\}_{u=0}^{\tau-1}$ and $\{\mathbf{q}_u\}_{u=0}^{\tau-1}$. The product $\hat{\mathbf{B}}_t^{-1}\mathbf{p} \;=\; \mathbf{q}_\tau$ is given by the last element of the latter sequence. Since determination of each of the elements of each sequence requires $O(n)$ operations and the total number of elements in each sequence is τ the total operation cost to compute both sequences is of order $O(\tau n)$. In computing $\hat{\mathbf{B}}_t^{-1}\mathbf{p}$ we also need to add the cost of the product $\mathbf{q}_0 = \hat{\mathbf{B}}_{t,0}^{-1}\mathbf{p}_\tau$ that links both sequences. To maintain overall computation cost of order $O(\tau n)$ this matrix has to have a sparse or low rank structure. A common choice in LBFGS, that we adopt for oLBFGS, is to make $\hat{\mathbf{B}}_{t,0}^{-1} = \hat{\gamma}_t\mathbf{I}$. The scalar constant $\hat{\gamma}_t$ is a function of the variable and stochastic gradient variations $\mathbf{v}_{t-1}$ and $\hat{\mathbf{r}}_{t-1}$, explicitly given by

$$\hat{\gamma}_t \;=\; \frac{\mathbf{v}_{t-1}^T\hat{\mathbf{r}}_{t-1}}{\hat{\mathbf{r}}_{t-1}^T\hat{\mathbf{r}}_{t-1}} \;=\; \frac{\mathbf{v}_{t-1}^T\hat{\mathbf{r}}_{t-1}}{\|\hat{\mathbf{r}}_{t-1}\|^2}\,. \tag{19}$$

with the value at the first iteration being $\hat{\gamma}_0 = 1$. The scaling factor $\hat{\gamma}_t$ attempts to estimate one of the eigenvalues of the Hessian matrix at step t and has been observed to work well in practice; see e.g., Liu and Nocedal (1989); Nocedal and Wright (1999). Further observe that the cost of computing $\hat{\gamma}_t$ is of order $O(n)$ and that since $\hat{\mathbf{B}}_{t,0}^{-1}$ is diagonal cost of computing the product $\mathbf{q}_0 = \hat{\mathbf{B}}_{t,0}^{-1}\mathbf{p}_\tau$ is also of order $O(n)$. We adopt the initialization in (19) in our subsequent analysis and numerical experiments.

The computation of the product $\hat{\mathbf{B}}_t^{-1}\mathbf{p}$ using the result in Proposition 1 is summarized in algorithmic form in the function in Algorithm 1. The function receives as arguments the initial matrix $\hat{\mathbf{B}}_{t,0}^{-1}$, the sequence of variable and stochastic gradient variations $\{\mathbf{v}_u, \hat{\mathbf{r}}_u\}_{u=t-\tau}^{t-1}$ and the vector $\mathbf{p}$ to produce the outcome $\mathbf{q} = \mathbf{q}_\tau = \hat{\mathbf{B}}_t^{-1}\mathbf{p}$. When called with the stochastic gradient $\mathbf{p} = \hat{\mathbf{s}}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t)$, the

Algorithm 1 Computation of oLBFGS step $\mathbf{q} = \hat{\mathbf{B}}_t^{-1}\mathbf{p}$ when called with $\mathbf{p} = \hat{\mathbf{s}}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t)$.

1: **function** $\mathbf{q} = \mathbf{q}_\tau$ = oLBFGS Step$\Big(\hat{\mathbf{B}}_{t,0}^{-1},\ \mathbf{p} = \mathbf{p}_0,\ \{\mathbf{v}_u, \hat{\mathbf{r}}_u\}_{u=t-\tau}^{t-1}\Big)$
2: **for** $u = 0, 1, \ldots, \tau - 1$ **do** {Loop to compute constants α_u and sequence $\mathbf{p}_u$}
3: Compute and store scalar $\alpha_u = \hat{\rho}_{t-u-1}\mathbf{v}_{t-u-1}^T\mathbf{p}_u$
4: Update sequence vector $\mathbf{p}_{u+1} = \mathbf{p}_u - \alpha_u\hat{\mathbf{r}}_{t-u-1}$. [cf. (17)]
5: **end for**
6: Multiply $\mathbf{p}_\tau$ by initial matrix: $\mathbf{q}_0 = \hat{\mathbf{B}}_{t,0}^{-1}\mathbf{p}_\tau$
7: **for** $u = 0, 1, \ldots, \tau - 1$ **do** {Loop to compute constants β_u and sequence $\mathbf{q}_u$}
8: Compute scalar $\beta_u = \hat{\rho}_{t-\tau+u}\hat{\mathbf{r}}_{t-\tau+u}^T\mathbf{q}_u$
9: Update sequence vector $\mathbf{q}_{u+1} = \mathbf{q}_u + (\alpha_{\tau-u-1} - \beta_u)\mathbf{v}_{t-\tau+u}$ [cf. (18)]
10: **end for** {return $\mathbf{q} = \mathbf{q}_\tau$}

function outputs the oLBFGS step $\hat{\mathbf{d}}_t := \hat{\mathbf{B}}_t^{-1}\hat{\mathbf{s}}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t)$ needed to implement the oLBFGS descent step in (10). The core of Algorithm 1 is given by the loop in steps 2-5 that computes the constants α_u and sequence elements $\mathbf{p}_u$ as well as the loop in steps 7-10 that computes the constants β_u and sequence elements $\mathbf{q}_u$. The two loops are linked by the initialization of the second sequence with the outcome of the first which is performed in Step 6. To implement the first loop we require τ inner products in Step 4 and τ vector summations in Step 5 which yield a total of $2\tau n$ multiplications. Likewise, the second loop requires τ inner products and τ vector summations in steps 9 and 10, respectively, which yields a total cost of also $2\tau n$ multiplications. Since the initial Hessian inverse approximation matrix $\hat{\mathbf{B}}_{t,0}^{-1}$ is diagonal the cost of computation $\hat{\mathbf{B}}_{t,0}^{-1}\mathbf{p}_\tau$ in Step 6 is n multiplications. Thus, Algorithm 1 requires a total of $(4\tau + 1)n$ multiplications which affirms the complexity cost of order $O(\tau n)$ for oLBFGS.

For reference, oLBFGS is also summarized in algorithmic form in Algorithm 2. As with any stochastic descent algorithm the descent iteration is implemented in three steps: the acquisition of L samples in Step 2, the computation of the stochastic gradient in Step 3, and the implementation of the descent update on the variable $\mathbf{w}_t$ in Step 6. Steps 4 and 5 are devoted to the computation of the oLBFGS descent direction $\hat{\mathbf{d}}_t$. In Step 4 we initialize the estimate $\hat{\mathbf{B}}_{t,0} = \hat{\gamma}_t\mathbf{I}$ as a scaled identity matrix using the expression for $\hat{\gamma}_t$ in (19) for $t > 0$. The value of $\gamma_t = \gamma_0$ for $t = 0$ is left as an input for the algorithm. We use $\hat{\gamma}_0 = 1$ in our numerical tests. In Step 5 we use Algorithm 1 for efficient computation of the descent direction $\hat{\mathbf{d}}_t = \hat{\mathbf{B}}_t^{-1}\hat{\mathbf{s}}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t)$. Step 7 determines the value of the stochastic gradient $\hat{\mathbf{s}}(\mathbf{w}_{t+1}, \tilde{\boldsymbol{\theta}}_t)$ so that the variable variations $\mathbf{v}_t$ and stochastic gradient variations $\hat{\mathbf{r}}_t$ become available for the computation of the curvature approximation matrix $\hat{\mathbf{B}}_t^{-1}$. In Step 8 the variable variation $\mathbf{v}_t$ and stochastic gradient variation $\hat{\mathbf{r}}_t$ are computed to be used in the next iteration. We analyze convergence properties of this algorithm in Section 3 and develop an application to search engine advertisement in Section 4.

3. Convergence Analysis

For the subsequent analysis it is convenient to define the instantaneous objective function associated with samples $\tilde{\boldsymbol{\theta}} = [\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_L]$ as

$$\hat{f}(\mathbf{w}, \tilde{\boldsymbol{\theta}}) := \frac{1}{L}\sum_{l=1}^{L} f(\mathbf{w}, \boldsymbol{\theta}_l). \tag{20}$$

The definition of the instantaneous objective function $\hat{f}(\mathbf{w}, \tilde{\boldsymbol{\theta}})$ in association with the fact that $F(\mathbf{w}) := \mathbb{E}_{\boldsymbol{\theta}}[f(\mathbf{w}, \boldsymbol{\theta})]$ implies that

$$F(\mathbf{w}) = \mathbb{E}_{\boldsymbol{\theta}}[\hat{f}(\mathbf{w}, \tilde{\boldsymbol{\theta}})]. \tag{21}$$

Algorithm 2 oLBFGS

Require: Initial value $\mathbf{w}_0$. Initial Hessian approximation parameter $\hat{\gamma}_0 = 1$.

for $t = 0, 1, 2, \ldots$ **do**

Acquire L independent samples $\tilde{\boldsymbol{\theta}}_t = [\boldsymbol{\theta}_{t1}, \ldots, \boldsymbol{\theta}_{tL}]$

Compute stochastic gradient: $\hat{\mathbf{s}}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t) = \dfrac{1}{L}\sum_{l=1}^{L} \nabla_{\mathbf{w}} f(\mathbf{w}_t, \boldsymbol{\theta}_{tl})$ [cf. (9)]

Initialize Hessian inverse estimate as $\hat{\mathbf{B}}_{t,0}^{-1} = \hat{\gamma}_t \mathbf{I}$ with $\hat{\gamma}_t = \dfrac{\mathbf{v}_{t-1}^T \hat{\mathbf{r}}_{t-1}}{\hat{\mathbf{r}}_{t-1}^T \hat{\mathbf{r}}_{t-1}}$ for $t > 0$ [cf (19)]

Compute descent direction with Algorithm 1: $\hat{\mathbf{d}}_t = \text{oLBFGS Step}\left(\hat{\mathbf{B}}_{t,0}^{-1},\ \hat{\mathbf{s}}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t),\ \{\mathbf{v}_u, \hat{\mathbf{r}}_u\}_{u=t-\tau}^{t-1}\right)$

Descend along direction $\hat{\mathbf{d}}_t$: $\mathbf{w}_{t+1} = \mathbf{w}_t - \epsilon_t \hat{\mathbf{d}}_t$ [cf. (10)]

Compute stochastic gradient: $\hat{\mathbf{s}}(\mathbf{w}_{t+1}, \tilde{\boldsymbol{\theta}}_t) = \dfrac{1}{L}\sum_{l=1}^{L} \nabla_{\mathbf{w}} f(\mathbf{w}_{t+1}, \boldsymbol{\theta}_{tl})$ [cf. (9)]

Variations $\mathbf{v}_t = \mathbf{w}_{t+1} - \mathbf{w}_t$ [variable, cf. (4)] $\hat{\mathbf{r}}_t = \hat{\mathbf{s}}(\mathbf{w}_{t+1}, \tilde{\boldsymbol{\theta}}_t) - \hat{\mathbf{s}}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t)$ [stoch. gradient, cf.(11)]

end for

Our goal here is to show that as time progresses the sequence of variable iterates $\mathbf{w}_t$ approaches the optimal argument $\mathbf{w}^*$. In proving this result we make the following assumptions.

Assumption 1 The instantaneous functions $\hat{f}(\mathbf{w}, \tilde{\boldsymbol{\theta}})$ are twice differentiable and the eigenvalues of the instantaneous Hessian $\hat{\mathbf{H}}(\mathbf{w}, \tilde{\boldsymbol{\theta}}) = \nabla_{\mathbf{w}}^2 \hat{f}(\mathbf{w}, \tilde{\boldsymbol{\theta}})$ are bounded between constants $0 < \tilde{m}$ and $\tilde{M} < \infty$ for all random variables $\tilde{\boldsymbol{\theta}}$,

$$\tilde{m}\mathbf{I} \preceq \hat{\mathbf{H}}(\mathbf{w}, \tilde{\boldsymbol{\theta}}) \preceq \tilde{M}\mathbf{I}. \tag{22}$$

Assumption 2 The second moment of the norm of the stochastic gradient is bounded for all $\mathbf{w}$. i.e., there exists a constant S^2 such that for all variables $\mathbf{w}$ it holds

$$\mathbb{E}_{\boldsymbol{\theta}}\left[\|\hat{\mathbf{s}}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t)\|^2 \,\middle|\, \mathbf{w}_t\right] \leq S^2. \tag{23}$$

Assumption 3 The step size sequence is selected as nonsummable but square summable, i.e.,

$$\sum_{t=0}^{\infty} \epsilon_t = \infty, \quad \text{and} \quad \sum_{t=0}^{\infty} \epsilon_t^2 < \infty. \tag{24}$$

Assumptions 2 and 3 are customary in stochastic optimization. The restriction imposed by Assumption 2 is intended to limit the random variation of stochastic gradients. If the variance of their norm is unbounded it is possible to have rare events that derail progress towards convergence. The condition in Assumption 3 balances descent towards optimal arguments – which requires a slowly decreasing step size – with the eventual elimination of random variations – which requires rapidly decreasing step sizes. An effective step size choice for which Assumption 3 holds is to make $\epsilon_t = \epsilon_0 T_0/(T_0 + t)$, for given parameters ϵ_0 and T_0 that control the initial step size and its speed of decrease, respectively. Assumption 1 is stronger than usual and specific to oLBFGS. Observe that considering the linearity of the expectation operator and the expression in (21) it follows that the Hessian of the average function can be written as $\nabla_{\mathbf{w}}^2 F(\mathbf{w}) = \mathbf{H}(\mathbf{w}) = \mathbb{E}_{\boldsymbol{\theta}}[\hat{\mathbf{H}}(\mathbf{w}, \tilde{\boldsymbol{\theta}})]$. Combining this observation with the bounds in (22) we conclude that there are constants $m \geq \tilde{m}$ and $M \leq \tilde{M}$ such that

$$\tilde{m}\mathbf{I} \preceq m\mathbf{I} \preceq \mathbf{H}(\mathbf{w}) \preceq M\mathbf{I} \preceq \tilde{M}\mathbf{I}. \tag{25}$$

The bounds in (25) are customary in convergence proofs of descent methods. For the results here the stronger condition spelled in Assumption 1 is needed. This assumption in necessary to guarantee that the inner product $\hat{\mathbf{r}}_t^T \mathbf{v}_t > 0$ is positive as we show in the following lemma.

Lemma 2 *Consider the stochastic gradient variation* $\hat{\mathbf{r}}_t$ *defined in* (11) *and the variable variation* $\mathbf{v}_t$ *defined in* (4)*. Let Assumption 1 hold so that we have lower and upper bounds* $\tilde{m}$ *and* $\tilde{M}$ *on the eigenvalues of the instantaneous Hessians. Then, for all steps* t *the inner product of variable and stochastic gradient variations* $\hat{\mathbf{r}}_t^T \mathbf{v}_t$ *is bounded below as*

$$\tilde{m}\|\mathbf{v}_t\|^2 \leq \hat{\mathbf{r}}_t^T \mathbf{v}_t \ . \tag{26}$$

Furthermore, the ratio of stochastic gradient variation squared norm $\|\hat{\mathbf{r}}_t\|^2 = \hat{\mathbf{r}}_t^T \hat{\mathbf{r}}_t$ *to inner product of variable and stochastic gradient variations is bounded as*

$$\tilde{m} \ \leq \ \frac{\hat{\mathbf{r}}_t^T \hat{\mathbf{r}}_t}{\hat{\mathbf{r}}_t^T \mathbf{v}_t} \ = \ \frac{\|\hat{\mathbf{r}}_t\|^2}{\hat{\mathbf{r}}_t^T \mathbf{v}_t} \ \leq \ \tilde{M}. \tag{27}$$

Proof See Appendix B. ■

According to Lemma 2, strong convexity of instantaneous functions $\hat{f}(\mathbf{w}, \tilde{\boldsymbol{\theta}})$ guaranties positiveness of the inner product $\mathbf{v}_t^T \hat{\mathbf{r}}_t$ as long as the variable variation is not identically null. In turn, this implies that the constant $\hat{\gamma}_t$ in (19) is nonnegative and that, as a consequence, the initial Hessian inverse approximation $\hat{\mathbf{B}}_{t,0}^{-1}$ is positive definite for all steps t. The positive definiteness of $\hat{\mathbf{B}}_{t,0}^{-1}$ in association with the positiveness of the inner product of variable and stochastic gradient variations $\mathbf{v}_t^T \hat{\mathbf{r}}_t > 0$ further guarantees that all the matrices $\hat{\mathbf{B}}_{t,u+1}^{-1}$, including the matrix $\hat{\mathbf{B}}_t^{-1} = \hat{\mathbf{B}}_{t,\tau}^{-1}$ in particular, that follow the update rule in (13) stay positive definite – see Mokhtari and Ribeiro (2014a) for details. This proves that (10) is a proper stochastic descent iteration because the stochastic gradient $\hat{\mathbf{s}}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t)$ is moderated by a positive definite matrix. However, this fact alone is not enough to guarantee convergence because the minimum and maximum eigenvalues of $\hat{\mathbf{B}}_t^{-1}$ could become arbitrarily small and arbitrarily large, respectively. To prove convergence we show this is not possible by deriving explicit lower and upper bounds on these eigenvalues.

The analysis is easier if we consider the matrix $\hat{\mathbf{B}}_t$ – as opposed to $\hat{\mathbf{B}}_t^{-1}$. Consider then the update in (13), and use the Sherman-Morrison formula to rewrite as an update that relates $\hat{\mathbf{B}}_{t,u+1}$ to $\hat{\mathbf{B}}_{t,u}$,

$$\hat{\mathbf{B}}_{t,u+1} \ = \ \hat{\mathbf{B}}_{t,u} - \frac{\hat{\mathbf{B}}_{t,u}\mathbf{v}_{t-\tau+u}\mathbf{v}_{t-\tau+u}^T\hat{\mathbf{B}}_{t,u}}{\mathbf{v}_{t-\tau+u}^T\hat{\mathbf{B}}_{t,u}\mathbf{v}_{t-\tau+u}} + \frac{\hat{\mathbf{r}}_{t-\tau+u}\hat{\mathbf{r}}_{t-\tau+u}^T}{\mathbf{v}_{t-\tau+u}^T\hat{\mathbf{r}}_{t-\tau+u}}, \tag{28}$$

for $u = 0, \ldots, \tau - 1$ and $\hat{\mathbf{B}}_{t,0} = 1/\hat{\gamma}_t \mathbf{I}$ as per (19). As in (13), the Hessian approximation at step t is $\hat{\mathbf{B}}_t = \hat{\mathbf{B}}_{t,\tau}$. In the following lemma we use the update formula in (28) to find bounds on the trace and determinant of the Hessian approximation $\hat{\mathbf{B}}_t$.

Lemma 3 *Consider the Hessian approximation* $\hat{\mathbf{B}}_t = \hat{\mathbf{B}}_{t,\tau}$ *defined by the recursion in* (28) *with* $\hat{\mathbf{B}}_{t,0} = \hat{\gamma}_t^{-1}\mathbf{I}$ *and* $\hat{\gamma}_t$ *as given by* (19)*. If Assumption 1 holds true, the trace* $\mathrm{tr}(\hat{\mathbf{B}}_t)$ *of the Hessian approximation* $\hat{\mathbf{B}}_t$ *is uniformly upper bounded for all times* $t \geq 1$*,*

$$\mathrm{tr}\left(\hat{\mathbf{B}}_t\right) \ \leq \ (n+\tau)\tilde{M}. \tag{29}$$

Likewise, if Assumption 1 holds true, the determinant $\det(\hat{\mathbf{B}}_t)$ *of the Hessian approximation* $\hat{\mathbf{B}}_t$ *is uniformly lower bounded for all times* t

$$\det\left(\hat{\mathbf{B}}_t\right) \ \geq \ \frac{\tilde{m}^{n+\tau}}{[(n+\tau)\tilde{M}]^{\tau}} \ . \tag{30}$$

Proof See Appendix C. ■

Lemma 3 states that the trace and determinants of the Hessian approximation matrix $\hat{\mathbf{B}}_t = \hat{\mathbf{B}}_{t,\tau}$ are bounded for all times $t \geq 1$. For time $t = 0$ we can write a similar bound that takes into account the fact that the constant γ_t that initializes the recursion in (28) is $\gamma_0 = 1$. Given that we are interested in an asymptotic convergence analysis, this bound in inconsequential. The bounds on the trace and determinant of $\hat{\mathbf{B}}_t$ are respectively equivalent to bounds in the sum and product of its eigenvalues. Further considering that the matrix $\hat{\mathbf{B}}_t$ is positive definite, as it follows from Lemma 2, these bounds can be further transformed into bounds on the smalls and largest eigenvalue of $\hat{\mathbf{B}}_t$. The resulting bounds are formally stated in the following lemma.

Lemma 4 *Consider the Hessian approximation $\hat{\mathbf{B}}_t = \hat{\mathbf{B}}_{t,\tau}$ defined by the recursion in (28) with $\hat{\mathbf{B}}_{t,0} = \hat{\gamma}_t^{-1}\mathbf{I}$ and $\hat{\gamma}_t$ as given by (19). Define the strictly positive constant $0 < c := \tilde{m}^{n+\tau}/[(n+\tau)\tilde{M}]^{n+\tau-1}$ and the finite constant $C := (n+\tau)\tilde{M} < \infty$. If Assumption 1 holds true, the range of eigenvalues of $\hat{\mathbf{B}}_t$ is bounded by c and C for all time steps $t \geq 1$, i.e.,*

$$\frac{\tilde{m}^{n+\tau}}{[(n+\tau)\tilde{M}]^{n+\tau-1}}\,\mathbf{I} \;=:\; c\mathbf{I} \;\preceq\; \hat{\mathbf{B}}_t \;\preceq\; C\mathbf{I} \;:=\; (n+\tau)\tilde{M}\,\mathbf{I}. \tag{31}$$

Proof See Appendix D. ∎

The bounds in Lemma 4 imply that their respective inverses are bounds on the range of the eigenvalues of the Hessian inverse approximation matrix $\hat{\mathbf{B}}_t^{-1}$. Specifically, the minimum eigenvalue of the Hessian inverse approximation $\hat{\mathbf{B}}_t^{-1}$ is larger than $1/C$ and the maximum eigenvalue of $\hat{\mathbf{B}}_t^{-1}$ does not exceed $1/c$, or, equivalently,

$$\frac{1}{C}\mathbf{I} \;\preceq\; \hat{\mathbf{B}}_t^{-1} \;\preceq\; \frac{1}{c}\mathbf{I}. \tag{32}$$

We further emphasize that the bounds in (32), or (31) for that matter, limit the conditioning of $\hat{\mathbf{B}}_t^{-1}$ for all realizations of the random samples $\{\tilde{\boldsymbol{\theta}}_t\}_{t=0}^{\infty}$, irrespective of the particular random draw. Having matrices $\hat{\mathbf{B}}_t^{-1}$ that are strictly positive definite with eigenvalues uniformly upper bounded by $1/c$ leads to the conclusion that if $\hat{\mathbf{s}}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t)$ is a descent direction, the same holds true of $\hat{\mathbf{B}}_t^{-1}\,\hat{\mathbf{s}}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t)$. The stochastic gradient $\hat{\mathbf{s}}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t)$ is not a descent direction in general, but we know that this is true for its conditional expectation $\mathbb{E}[\hat{\mathbf{s}}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t) \,\big|\, \mathbf{w}_t] = \nabla F(\mathbf{w}_t)$. Hence, we conclude that $\hat{\mathbf{B}}_t^{-1}\hat{\mathbf{s}}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t)$ is an average descent direction since $\mathbb{E}[\hat{\mathbf{B}}_t^{-1}\hat{\mathbf{s}}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t)\big|\mathbf{w}_t] = \hat{\mathbf{B}}_t^{-1}\nabla F(\mathbf{w}_t)$. Stochastic optimization methods whose displacements $\mathbf{w}_{t+1} - \mathbf{w}_t$ are descent directions on average are expected to approach optimal arguments. We show that this is true of oLBFGS in the following lemma.

Lemma 5 *Consider the online Limited memory BFGS algorithm as defined by the descent iteration in (10) with matrices $\hat{\mathbf{B}}_t^{-1} = \hat{\mathbf{B}}_{t,\tau}^{-1}$ obtained after τ recursive applications of the update in (13) initialized with $\hat{\mathbf{B}}_{t,0}^{-1} = \hat{\gamma}_t\mathbf{I}$ and $\hat{\gamma}_t$ as given by (19). If Assumptions 1 and 2 hold true, the sequence of average function values $F(\mathbf{w}_t)$ satisfies*

$$\mathbb{E}\left[F(\mathbf{w}_{t+1}) \,\middle|\, \mathbf{w}_t\right] - F(\mathbf{w}^*) \leq F(\mathbf{w}_t) - F(\mathbf{w}^*) - \frac{\epsilon_t}{C}\|\nabla F(\mathbf{w}_t)\|^2 + \frac{MS^2\epsilon_t^2}{2c^2}. \tag{33}$$

Proof See Appendix E. ∎

Setting aside the term $MS^2\epsilon_t^2/2c^2$ for the sake of argument, (88) defines a supermartingale relationship for the sequence of objective function errors $F(\mathbf{w}_t) - F(\mathbf{w}^*)$. This implies that the sequence $\epsilon_t\|\nabla F(\mathbf{w}_t)\|^2/C$ is almost surely summable which, given that the step sizes ϵ_t are nonsummable as per (24), further implies that the limit infimum $\liminf_{t\to\infty}\|\nabla F(\mathbf{w}_t)\|$ of the gradient norm

$\|\nabla F(\mathbf{w}_t)\|$ is almost surely null. This latter observation is equivalent to having $\liminf_{t\to\infty} F(\mathbf{w}_t) - F(\mathbf{w}^*) = 0$ with probability 1 over realizations of the random samples $\{\tilde{\boldsymbol{\theta}}_t\}_{t=0}^{\infty}$. Therefore, a subsequence of the sequence of objective function errors $F(\mathbf{w}_t) - F(\mathbf{w}^*)$ converges to null almost surely. Moreover, according to the result of supermartingale convergence theorem, the limit $\lim_{t\to\infty} F(\mathbf{w}_t) - F(\mathbf{w}^*)$ of the nonnegative objective function errors $F(\mathbf{w}_t) - F(\mathbf{w}^*)$ almost surely exists. This observation in conjunction with the fact that a subsequence of the sequence $F(\mathbf{w}_t) - F(\mathbf{w}^*)$ converges almost surely to null implies that the whole sequence of $F(\mathbf{w}_t) - F(\mathbf{w}^*)$ converges almost surely to zero. Considering the strong convexity assumption this result implies almost sure convergence of the whole sequence of $\|\mathbf{w}_t - \mathbf{w}^*\|^2$ to null. The term $MS^2\epsilon_t^2/2c^2$ is a relatively minor nuisance that can be taken care of with a technical argument that we present in the proof of the following theorem.

Theorem 6 *Consider the online Limited memory BFGS algorithm as defined by the descent iteration in* (10) *with matrices* $\hat{\mathbf{B}}_t^{-1} = \hat{\mathbf{B}}_{t,\tau}^{-1}$ *obtained after* τ *recursive applications of the update in* (13) *initialized with* $\hat{\mathbf{B}}_{t,0}^{-1} = \hat{\gamma}_t \mathbf{I}$ *and* $\hat{\gamma}_t$ *as given by* (19). *If Assumptions 1-3 hold true the limit of the squared Euclidean distance to optimality* $\|\mathbf{w}_t - \mathbf{w}^*\|^2$ *converges to zero almost surely, i.e.,*

$$\Pr\left[\lim_{t\to\infty} \|\mathbf{w}_t - \mathbf{w}^*\|^2 = 0\right] = 1, \tag{34}$$

where the probability is over realizations of the random samples $\{\tilde{\boldsymbol{\theta}}_t\}_{t=0}^{\infty}$.

Proof See Appendix F. ■

Theorem 6 establishes convergence of the oLBFGS algorithm summarized in Algorithm 2. The lower and upper bounds on the eigenvalues of $\hat{\mathbf{B}}_t$ derived in Lemma 4 play a fundamental role in the proofs of the prerequisite Lemma 5 and Theorem 6 proper. Roughly speaking, the lower bound on the eigenvalues of $\hat{\mathbf{B}}_t$ results in an upper bound on the eigenvalues of $\hat{\mathbf{B}}_t^{-1}$ which limits the effect of random variations on the stochastic gradient $\hat{\mathbf{s}}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t)$. If this bound does not exist – as is the case, e.g., of regular stochastic BFGS – we may observe catastrophic amplification of random variations of the stochastic gradient. The upper bound on the eigenvalues of $\hat{\mathbf{B}}_t$, which results in a lower bound on the eigenvalues of $\hat{\mathbf{B}}_t^{-1}$, guarantees that the random variations in the curvature estimate $\hat{\mathbf{B}}_t$ do not yield matrices with arbitrarily small norm. If this bound does not hold, it is possible to end up halting progress before convergence as the stochastic gradient is nullified by multiplication with an arbitrarily small eigenvalue.

The result in Theorem 6 is strong because it holds almost surely over realizations of the random samples $\{\tilde{\boldsymbol{\theta}}_t\}_{t=0}^{\infty}$ but not stronger than the same convergence guarantees that hold for SGD. We complement the convergence result in Theorem 6 with a characterization of the expected convergence rate that we introduce in the following theorem.

Theorem 7 *Consider the online Limited memory BFGS algorithm as defined by the descent iteration in* (10) *with matrices* $\hat{\mathbf{B}}_t^{-1} = \hat{\mathbf{B}}_{t,\tau}^{-1}$ *obtained after* τ *recursive applications of the update in* (13) *initialized with* $\hat{\mathbf{B}}_{t,0}^{-1} = \hat{\gamma}_t \mathbf{I}$ *and* $\hat{\gamma}_t$ *as given by* (19). *Let Assumptions 1 and 2 hold, and further assume that the step size sequence is of the form* $\epsilon_t = \epsilon_0/(t + T_0)$ *with the parameters* ϵ_0 *and* T_0 *satisfying the inequality* $2m\epsilon_0 T_0/C > 1$. *Then, the difference between the expected optimal objective* $\mathbb{E}[F(\mathbf{w}_t)]$ *and the optimal objective* $F(\mathbf{w}^*)$ *is bounded as*

$$\mathbb{E}[F(\mathbf{w}_t)] - F(\mathbf{w}^*) \;\leq\; \frac{C_0}{T_0 + t}\;, \tag{35}$$

where the constant C_0 *is defined as*

$$C_0 := \max\left\{\frac{\epsilon_0^2\, T_0^2 CMS^2}{2c^2(2m\epsilon_0 T_0 - C)}\;,\; T_0\left(F(\mathbf{w}_0) - F(\mathbf{w}^*)\right)\right\}. \tag{36}$$

Proof See Appendix G. ■

Theorem 7 shows that under specified assumptions the expected error in terms of the objective value after t oLBFGS iterations is of order $O(1/t)$. As is the case of Theorem 6, this result is not better than the convergence rate of conventional SGD. As can be seen in the proof of Theorem 7, the convergence rate is dominated by the noise term introduced by the difference between stochastic and regular gradients. This noise term would be present even if exact Hessians were available and in that sense the best that can be proven of oLBFGS is that the convergence rate is not worse than that of SGD. Given that theorems 6 and 7 parallel the theoretical guarantees of SGD it is perhaps fairer to describe oLBFGS as an adaptive reconditioning strategy instead of a stochastic quasi-Newton method. The latter description refers to the genesis of the algorithm, but the former is more accurate description of its behavior. Do notice that while the convergence rate doesn't change, improvements in convergence time are significant as we illustrate with the numerical experiments that we present in the next section.

4. Search Engine Advertising

We apply oLBFGS to the problem of predicting the click-through rate (CTR) of an advertisement displayed in response to a specific search engine query by a specific visitor. In these problems we are given meta information about an advertisement, the words that appear in the query, as well as some information about the visitor and are asked to predict the likelihood that this particular ad is clicked by this particular user when performing this particular query. The information specific to the ad includes descriptors of different characteristics such as the words that appear in the title, the name of the advertiser, keywords that identify the product, and the position on the page where the ad is to be displayed. The information specific to the user is also heterogeneous and includes gender, age, and propensity to click on ads. To train a classifier we are given information about past queries along with the corresponding click success of the ads displayed in response to the query. The ad metadata along with user data and search words define a feature vector that we use to train a logistic regressor that predicts the CTR of future ads. Given the heterogeneity of the components of the feature vector we expect a logistic cost function with skewed level sets and consequent large benefits from the use of oLBFGS.

4.1 Feature Vectors

For the CTR problem considered here we use the Tencent search engine data set Sun (2012). This data set contains the outcomes of 236 million (236×10^6) searches along with information about the ad, the query, and the user. The information contained in each sample point is the following:

- User profile: If known, age and gender of visitor performing query.
- Depth: Total number of advertisements displayed in the search results page.
- Position: Position of the advertisement in the search page.
- Impression: Number of times the ad was displayed to the user who issued the query.
- Query: The words that appear in the user's query.
- Title: The words that appear in the title of ad.
- Keywords: Selected keywords that specify the type of product.
- Ad ID: Unique identifier assigned to each specific advertisement.

		Nonzero components	
Feature type	**Total components**	**Maximum (observed/structure)**	**Mean (observed)**
Age	6	1 (structure)	1.0
Gender	3	1 (structure)	1.0
Impression	3	1 (structure)	1.0
Depth	3	1 (structure)	1.0
Position	3	1 (structure)	1.0
Query	20,000	125 (observed)	3.0
Title	20,000	29 (observed)	8.8
Keyword	20,000	16 (observed)	2.1
Advertiser ID	5,184	1 (structure)	1.0
Advertisement ID	108,824	1 (structure)	1.0
Total	**174,026**	**148 (observed)**	**20.9**

Table 1: Components of the feature vector for prediction of advertisements click-through rates. For each feature class we report the total number of components in the feature vector as well as the maximum and average number of nonzero components.

- Advertiser ID: Unique identifier assigned to each specific advertiser.
- Clicks: Number of times the user clicked on the ad.

From this information we create a set of feature vectors $\{\mathbf{x}_i\}_{i=1}^N$, with corresponding labels $y_i \in \{-1, 1\}$. The label associated with feature vector $\mathbf{x}_i$ is $y_i = 1$ if the number of clicks in the ad is more than 0. Otherwise the label is $y_i = -1$. We use a binary encoding for all the features in the vector $\mathbf{x}_i$. For the age of the user we use the six age intervals $(0, 12]$, $(12, 18]$, $(18, 24]$, $(24, 30]$, $(30, 40]$, and $(40, \infty)$ to construct six indicator entries in $\mathbf{x}_i$ that take the value 1 if the age of the user is known to be in the corresponding interval. E.g., a 21 year old user has an age that falls in the third interval which implies that we make $[\mathbf{x}_i]_3 = 1$ and $[\mathbf{x}_i]_k = 0$ for all other k between 1 and 6. If the age of the user is unknown we make $[\mathbf{x}_i]_k = 0$ for all k between 1 and 6. For the gender of the visitors we use the next three components of $\mathbf{x}_i$ to indicate male, female, or unknown gender. For a male user we make $[\mathbf{x}_i]_7 = 1$, for a female user $[\mathbf{x}_i]_8 = 1$, and for visitors of unknown gender we make $[\mathbf{x}_i]_9 = 1$. The next three components of $\mathbf{x}_i$ are used for the depth feature. If the the number of advertisements displayed in the search page is 1 we make $[\mathbf{x}_i]_{10} = 1$, if 2 different ads are shown we make $[\mathbf{x}_i]_{11} = 1$, and for depths of 3 or more we make $[\mathbf{x}_i]_{12} = 1$. To indicate the position of the ad in the search page we also use three components of $\mathbf{x}_i$. We use $[\mathbf{x}_i]_{13} = 1$, $[\mathbf{x}_i]_{14} = 1$, and $[\mathbf{x}_i]_{15} = 1$ to indicate that the ad is displayed in the first, second, and third position, respectively. Likewise we use $[\mathbf{x}_i]_{16}$, $[\mathbf{x}_i]_{17}$ and $[\mathbf{x}_i]_{18}$ to indicate that the impression of the ad is 1, 2 or more than 3.

For the words that appear in the query we have in the order of 10^5 distinct words. To reduce the number of elements necessary for this encoding we create 20,000 bags of words through random hashing with each bag containing 5 or 6 distinct words. Each of these bags is assigned an index k. For each of the words in the query we find the bag in which this word appears. If the word appears in the kth bag we indicate this occurrence by setting the $k + 18$th component of the feature vector to $[\mathbf{x}_i]_{k+18} = 1$. Observe that since we use 20,000 bags, components 19 through 20,018 of $\mathbf{x}_i$ indicate the presence of specific words in the query. Further note that we may have more than one $\mathbf{x}_i$ component different from zero because there may be many words in the query, but that the total number of nonzero elements is much smaller than 20,000. On average, 3.0 of these elements of

the feature vector are nonzero. The same bags of words are used to encode the words that appear in the title of the ad and the product keywords. We encode the words that appear in the title of the ad by using the next $20,000$ components of vector $\mathbf{x}_i$, i.e. components $20,019$ through $40,018$. Components $40,019$ through $60,018$ are used to encode product keywords. As in the case of the words in the search just a few of these components are nonzero. On average, the number of nonzero components of feature vectors that describe the title features is 8.8. For product keywords the average is 2.1. Since the number of distinct advertisers in the training set is $5,184$ we use feature components $60,019$ through 65202 to encode this information. For the kth advertiser ID we set the $k + 60,018th$ component of the feature vector to $[\mathbf{x}_i]_{k+60,018} = 1$. Since the number of distinct advertisements is $108,824$ we allocate the last $108,824$ components of the feature vector to encode the ad ID. Observe that only one out of $5,184$ advertiser ID components and one of the $108,824$ advertisement ID components are nonzero.

In total, the length of the feature vector is 174,026 where each of the components are either 0 or 1. The vector is very sparse. We observe a maximum of 148 nonzero elements and an average of 20.9 nonzero elements in the training set – see Table 1. This is important because the cost of implementing inner products in the oLBFGS training of the logistic regressor that we introduce in the following section is proportional to the number of nonzero elements in $\mathbf{x}_i$.

4.2 Logistic Regression of Click-Through Rate

We use the training set to estimate the CTR with a logistic regression. For that purpose let $\mathbf{x} \in \mathbb{R}^n$ be a vector containing the features described in Section 4.1, $\mathbf{w} \in \mathbb{R}^n$ a classifier that we want to train, and $y \in -1, 1$ an indicator variable that takes the value $y = 1$ when the ad presented to the user is clicked and $y = -1$ when the ad is not clicked by the user. We hypothesize that the CTR, defined as the probability of observing $y = 1$, can be written as the logistic function

$$\text{CTR}(\mathbf{x}; \mathbf{w}) \ := \text{P}\left[y = 1 \,\middle|\, \mathbf{x}; \mathbf{w}\right] \ = \ \frac{1}{1 + \exp\left(-\mathbf{x}^T\mathbf{w}\right)} \ . \tag{37}$$

We read (37) as stating that for a feature vector $\mathbf{x}$ the CTR is determined by the inner product $\mathbf{x}^T\mathbf{w}$ through the given logistic transformation.

Consider now the training set $\mathcal{S} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ which contains N realizations of features $\mathbf{x}_i$ and respective click outcomes y_i and further define the sets $\mathcal{S}_1 := \{(\mathbf{x}_i, y_i) \in \mathcal{S} : y_i = 1\}$ and $\mathcal{S}_{-1} := \{(\mathbf{x}_i, y_i) \in \mathcal{S} : y_i = -1\}$ containing clicked and unclicked advertisements, respectively. With the data given in $\mathcal{S}$ we define the optimal classifier $\mathbf{w}^*$ as a maximum likelihood estimate (MLE) of $\mathbf{w}$ given the model in (37) and the training set $\mathcal{S}$. This MLE can be found as the minimizer of the log-likelihood loss

$$\begin{aligned}
\mathbf{w}^* \ &:= \ \text{argmin} \, \frac{\lambda}{2}\|\mathbf{w}\|^2 \ + \ \frac{1}{N}\sum_{i=1}^{N} \log\Big(1 + \exp\big(-y_i\mathbf{x}_i^T\mathbf{w}\big)\Big) \\
&= \ \text{argmin} \, \frac{\lambda}{2}\|\mathbf{w}\|^2 \ + \ \frac{1}{N}\Bigg[\sum_{\mathbf{x}_i \in \mathcal{S}_1} \log\Big(1 + \exp(-\mathbf{x}_i^T\mathbf{w})\Big) + \sum_{\mathbf{x}_i \in \mathcal{S}_{-1}} \log\Big(1 + \exp(\mathbf{x}_i^T\mathbf{w})\Big)\Bigg],
\end{aligned} \tag{38}$$

where we have added the regularization term $\lambda\|\mathbf{w}\|^2/2$ to disincentivize large values in the weight vector $\mathbf{w}^*$; see e.g., Ng (2004).

The practical use of (37) and (38) is as follows. We use the data collected in the training set $\mathcal{S}$ to determine the vector $\mathbf{w}^*$ in (38). When a user issues a query we concatenate the user and query specific elements of the feature vector with the ad specific elements of several candidate ads. We then proceed to display the advertisement with, say, the largest CTR. We can interpret the set $\mathcal{S}$ as having been acquired offline or online. In the former case we want to use a stochastic optimization algorithm because computing gradients is infeasible – recall that we are considering training samples

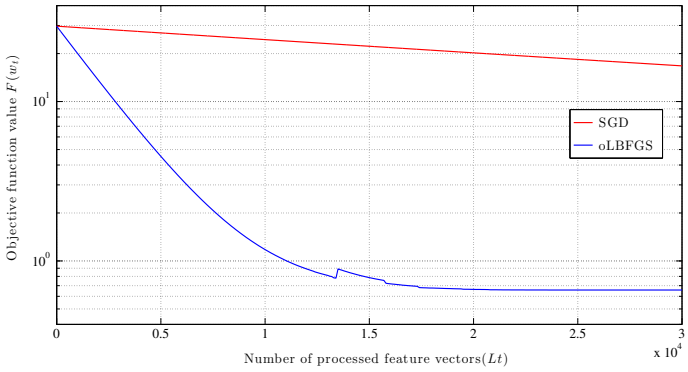


Figure 1: Illustration of Negative log-likelihood value for oLBFGS and SGD after processing certain amount of feature vectors. The accuracy of oLBFGS is better than SGD after processing a specific number of feature vectors.

with a number of elements N in the order of 10^6. The performance metric of interest in this case is the logistic cost as a function of computational time. If elements of $\mathcal{S}$ are acquired online we update $\mathbf{w}$ whenever a new vector becomes available so as to adapt to changes in preferences. In this case we want to exploit the information in new samples as much as possible. The correct metric in this case is the logistic cost as a function of the number of feature vectors processed. We use the latter metric for the numerical experiments in the following section.

4.3 Numerical Results

Out of the 236×10^6 in the Tencent data set we select 10^6 sample points to use as the training set $\mathcal{S}$ and 10^5 sample points to use as a test set $\mathcal{T}$. To select elements of the training and test set we divide the first 1.1×10^6 sample points of the complete data set in 10^5 consecutive blocks with 11 elements. The first 10 elements of the block are assigned to the training set and the 11th element to the test set. To solve for the optimal classifier we implement SGD and oLBFGS by selecting feature vectors $\mathbf{x}_i$ at random from the training set $\mathcal{S}$. In all of our numerical experiments the regularization parameter in (38) is $\lambda = 10^{-6}$. The step sizes for both algorithms are of the form $\epsilon_t = \epsilon_0 T_0/(T_0 + t)$. We set $\epsilon_0 = 10^{-2}$ and $T_0 = 10^4$ for oLBFGS and $\epsilon_0 = 10^{-1}$ and $T_0 = 10^6$ for SGD. For SGD the sample size in (9) is set to $L = 20$ whereas for oLBFGS it is set to $L = 100$. The values of parameters ϵ_0, T_0, and L are chosen to yield best convergence times in a rough parameter optimization search. Observe the relatively large values of L that are used to compute stochastic gradients. This is necessary due to the extreme sparsity of the feature vectors $\mathbf{x}_i$ that contain an average of only 20.9 nonzero out 174,026 elements. Even when considering $L = 100$ vectors they are close to orthogonal. The size of memory for oLBFGS is set to $\tau = 10$. With $L = 100$ features with an average sparsity of 20.9 nonzero elements and memory $\tau = 10$ the cost of each oLBFGS iteration is in the order of 2.1×10^4 operations.

Figure 1 illustrates the convergence path of SGD and oLBFGS on the advertising training set. We depict the value of the log likelihood objective in (38) evaluated at $\mathbf{w} = \mathbf{w}_t$ where $\mathbf{w}_t$ is the classifier iterate determined by SGD or oLBFGS. The horizontal axis is scaled by the number of feature vectors L that are used in the evaluation of stochastic gradients. This results in a plot of log likelihood cost versus the number Lt of feature vectors processed. To read iteration indexes from Figure 1 divide the horizontal axis values by $L = 100$ for oLBFGS and $L = 20$ for SGD. The curvature correction of oLBFGS results in significant reductions in convergence time. For way of

illustration observe that after processing $Lt = 3 \times 10^4$ feature vectors the objective value achieved by oLBFGS is $F(\mathbf{w}_t) = 0.65$, while for SGD it still stands at $F(\mathbf{w}_t) = 16$ which is a meager reduction from the random initialization point at which $F(\mathbf{w}_0) = 30$. In fact, oLBFGS converges to the minimum possible log likelihood cost $F(\mathbf{w}_t) = 0.65$ after processing 1.7×10^4 feature vectors. This illustration hints that oLBFGS makes better use of the information available in feature vectors.

To corroborate that the advantage of oLBFGS is not just an artifact of the structure of the log likelihood cost in (38) we process 2×10^4 feature vectors with SGD and oLBFGS and evaluate the predictive accuracy of the respective classifiers on the test set. As measures of predictive accuracy we adopt the frequency histogram of the predicted click through rate $\mathrm{CTR}(\mathbf{x}; \mathbf{w})$ for all clicked ads and the frequency histogram of the complementary predicted click through rate $1 - \mathrm{CTR}(\mathbf{x}; \mathbf{w})$ for all the ads that were *not* clicked. To do so we separate the test set by defining the set $\mathcal{T}_1 := \{(\mathbf{x}_i, y_i) \in \mathcal{T} : y_i = 1\}$ of clicked ads and the set $\mathcal{T}_{-1} := \{(\mathbf{x}_i, y_i) \in \mathcal{T} : y_i = -1\}$ of ads in the test set that were not clicked. For a given classifier $\mathbf{w}$ we compute the predicted probability $\mathrm{CTR}(\mathbf{x}_i; \mathbf{w})$ for each of the ads in the clicked set $\mathcal{T}_1$. We then consider a given interval $[a, b]$ and define the frequency histogram of the predicted click through rate as the fraction of clicked ads for which the prediction $\mathrm{CTR}(\mathbf{x}_i; \mathbf{w})$ falls in $[a, b]$,

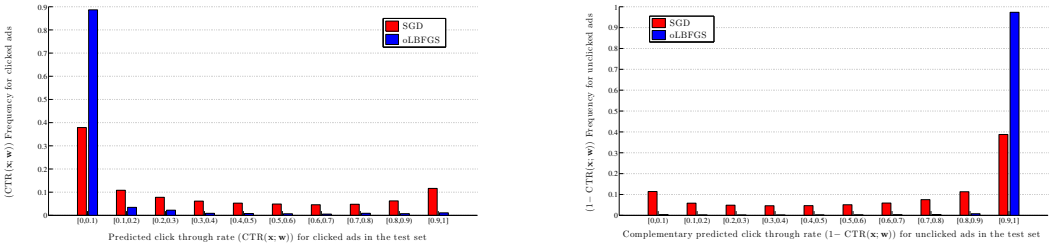
$$\mathcal{H}_1(\mathbf{w}; a, b) := \frac{1}{\#(\mathcal{T}_1)} \sum_{(\mathbf{x}_i, y_i) \in \mathcal{T}_1} \mathbb{I}\Big\{\mathrm{CTR}(\mathbf{x}_i; \mathbf{w}) \in [a, b]\Big\}, \tag{39}$$

where $\#(\mathcal{T}_1)$ denotes the cardinality of the set $\mathcal{T}_1$. Likewise, we consider the ads in the set $\mathcal{T}_{-1}$ that were not clicked and compute the prediction $1 - \mathrm{CTR}(\mathbf{x}_i; \mathbf{w})$ on the probability of the ad not being clicked. We then consider a given interval $[a, b]$ and define the frequency histogram $\mathcal{H}_{-1}(\mathbf{w}; a, b)$ as the fraction of unclicked ads for which the prediction $1 - \mathrm{CTR}(\mathbf{x}_i; \mathbf{w})$ falls in $[a, b]$,

$$\mathcal{H}_{-1}(\mathbf{w}; a, b) := \frac{1}{\#(\mathcal{T}_{-1})} \sum_{(\mathbf{x}_i, y_i) \in \mathcal{T}_{-1}} \mathbb{I}\Big\{1 - \mathrm{CTR}(\mathbf{x}_i; \mathbf{w}) \in [a, b]\Big\}. \tag{40}$$

The histogram $\mathcal{H}_1(\mathbf{w}; a, b)$ in (39) allows us to study how large the predicted probability $\mathrm{CTR}(\mathbf{x}_i; \mathbf{w})$ is for the clicked ads. Conversely, the histogram $\mathcal{H}_{-1}(\mathbf{w}; a, b)$ in (40) gives an indication of how large the predicted probability $1 - \mathrm{CTR}(\mathbf{x}_i; \mathbf{w})$ is for the unclicked ads. An ideal classifier is one for which the frequency counts in $\mathcal{H}_1(\mathbf{w}; a, b)$ accumulate at $\mathrm{CTR}(\mathbf{x}_i; \mathbf{w}) = 1$ and for which $\mathcal{H}_{-1}(\mathbf{w}; a, b)$ accumulates observations at $1 - \mathrm{CTR}(\mathbf{x}_i; \mathbf{w}) = 1$. This corresponds to a classifier that predicts a click probability of 1 for all ads that were clicked and a click probability of 0 for all ads that were not clicked.

Fig. 2(a) shows the histograms of predicted click through rate $\mathrm{CTR}(\mathbf{x}; \mathbf{w})$ for all clicked ads by oLBFGS and SGD classifiers after processing 2×10^4 training sample points. oLBFGS classifier for 88% of test points in $\mathcal{T}_1$ predicts $\mathrm{CTR}(\mathbf{x}; \mathbf{w})$ in the interval $[0, 0.1]$ and the classifier computed by SGD estimates the click through rate $\mathrm{CTR}(\mathbf{x}; \mathbf{w})$ in the same interval for 37% of clicked ads in the test set. These numbers shows the inaccurate click through rate predictions of both classifiers for the test points with label $y = 1$. Although, SGD and oLBFGS classifiers have catastrophic performances in predicting click through rate $\mathrm{CTR}(\mathbf{x}; \mathbf{w})$ for the clicked ads in the test set, they perform well in estimating complementary predicted click through rate $1 - \mathrm{CTR}(\mathbf{x}; \mathbf{w})$ for the test points with label $y = -1$. This observation implied by Fig. 2(b) which shows the histograms of complementary predicted click through rate $1 - \mathrm{CTR}(\mathbf{x}; \mathbf{w})$ for all *not* clicked ads by oLBFGS and SGD classifiers after processing 2×10^4 training sample points. As it shows after processing 2×10^4 sample points of the training set the predicted probability $1 - \mathrm{CTR}(\mathbf{x}; \mathbf{w})$ by the SGD classifier for 38.8% of the test points are in the interval $[0.9, 1]$, while for the classifier computed by oLBFGS 97.3% of predicted probability $1 - \mathrm{CTR}(\mathbf{x}; \mathbf{w})$ are in the interval $[0.9, 1]$ which is a significant performance.



(a) Histogram $\mathcal{H}_1(\mathbf{w}; a, b)$, [cf. (39)].

(b) Histogram $\mathcal{H}_{-1}(\mathbf{w}; a, b)$, [cf. (40)].

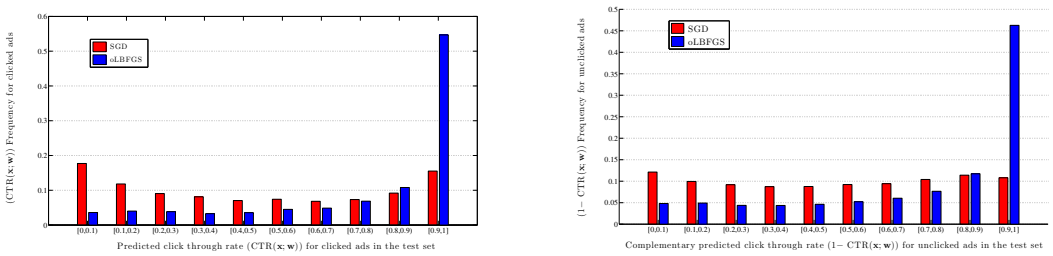
Figure 2: Performance of classifier after processing 2×10^4 feature vectors with SGD and oLBFGS for the cost in (38). Histograms for: (a) predicted click through rate $\text{CTR}(\mathbf{x}; \mathbf{w})$ for all clicked ads; and (b) complementary predicted click through rate $1 - \text{CTR}(\mathbf{x}; \mathbf{w})$ for all unclicked ads. For an ideal classifier that predicts a click probability $\text{CTR}(\mathbf{x}; \mathbf{w}) = 1$ for all clicked ads and a click probability $\text{CTR}(\mathbf{x}; \mathbf{w}) = 0$ for all unclicked ads the frequency counts in $\mathcal{H}_1(\mathbf{w}; a, b)$ and $\mathcal{H}_{-1}(\mathbf{w}; a, b)$ would accumulate in the $[0.9, 1]$ bin. Neither SGD nor oLBFGS compute acceptable classifiers because the number of clicked ads in the test set is very small and predicting $\text{CTR}(\mathbf{x}; \mathbf{w}) = 0$ for all ads is close to the minimum of (38).

The reason for the inaccurate predictions of both classifiers is that most elements in the training set $\mathcal{S}$ are unclicked ads. Thus, the minimizer $\mathbf{w}^*$ of the log likelihood cost in (38) is close to a classifier that predicts $\text{CTR}(\mathbf{x}; \mathbf{w}^*) \approx 0$ for most ads. Indeed, out of the 10^6 elements in the training set, 94.8% of them have labels $y_i = -1$ and only the remaining 5.2×10^4 feature vectors correspond to clicked ads. To overcome this problem we replicate observations with labels $y_i = 1$ to balance the representation of both labels in the training set. Equivalently, we introduce a constant γ and redefine the log likelihood objective in (38) to give a larger weight to feature vectors that correspond to clicked ads,

$$\mathbf{w}^* \;=\; \text{argmin}\; \frac{\lambda}{2}\|\mathbf{w}\|^2 \;+\; \frac{1}{M}\Bigg[\; \gamma \sum_{\mathbf{x}_i \in \mathcal{S}_1} \log\Big(1 + \exp(-\mathbf{x}_i^T \mathbf{w})\Big) + \sum_{\mathbf{x}_i \in \mathcal{S}_{-1}} \log\Big(1 + \exp(\mathbf{x}_i^T \mathbf{w})\Big)\Bigg], \quad (41)$$

where we defined $M := \gamma \#(\mathcal{S}_1) + \#(\mathcal{S}_{-1})$ to account for the replication of clicked featured vectors that is implicit in (41). To implement SGD and oLBFGS in the weighted log function in (41) we need to bias the random choice of feature vector so that vectors in $\mathcal{S}_1$ are γ times more likely to be selected than vectors in $\mathcal{S}_2$. Although our justification to introduce γ is to balance the types of feature vectors, γ is just a tradeoff constant to increase the percentage of correct predictions for clicked ads – which is close to zero in Figure 2 – at the cost of reducing the accuracy of correct predictions of unclicked ads – which is close to one in Figure 2.

We repeat the experiment of processing 2×10^4 feature vectors that we summarized in Figure 2 but now we use the objective cost in (41) instead of the cost in (38). We set $\gamma = 18.2$ which makes replicated clicked ads as numerous as unclicked ads. The resulting SGD and oLBFGS histograms of the predicted click through rates for all clicked ads and complementary predicted click through rates for all unclicked ads are shown in Figure 3. In particular, Figure 3(a) shows the histograms of predicted click through rate $\text{CTR}(\mathbf{x}; \mathbf{w})$ for all clicked ads after processing 2×10^4 training sample points. The modification of the log likelihood cost increases the accuracy of the oLBFGS classifier which is now predicting a click probability $\text{CTR}(\mathbf{x}; \mathbf{w}) \in [0.9, 1]$ for 54.7% of the ads that were indeed clicked. There is also improvement for the SGD classifier but the prediction is much less impressive. Only 15.5% of the clicked ads are associated with a click probability prediction in the interval $[0.9, 1]$. This improvement is at the cost of reducing the complementary predicted click through



(a) Histogram $\mathcal{H}_1(\mathbf{w}; a, b)$, [cf. (39)].

(b) Histogram $\mathcal{H}_{-1}(\mathbf{w}; a, b)$, [cf. (40)].

Figure 3: Performance of classifier after processing 2×10^4 feature vectors with SGD and oLBFGS for the cost in (41). Histograms for: (a) predicted click through rate $\text{CTR}(\mathbf{x}; \mathbf{w})$ for all clicked ads; and (b) complementary predicted click through rate $1 - \text{CTR}(\mathbf{x}; \mathbf{w})$ for all unclicked ads. For an ideal classifier that predicts a click probability $\text{CTR}(\mathbf{x}; \mathbf{w}) = 1$ for all clicked ads and a click probability $\text{CTR}(\mathbf{x}; \mathbf{w}) = 0$ for all unclicked ads the frequency counts in $\mathcal{H}_1(\mathbf{w}; a, b)$ and $\mathcal{H}_{-1}(\mathbf{w}; a, b)$ would accumulate in the $[0.9, 1]$ bin. The classifier computed by oLBFGS is much more accurate than the one computed by SGD.

rate $1 - \text{CTR}(\mathbf{x}; \mathbf{w})$ for the ads that were indeed not clicked. However, the classifier computed by oLBFGS after processing 2×10^4 feature vectors still predicts a probability $1 - \text{CTR}(\mathbf{x}; \mathbf{w}) \in [0.9, 1]$ for 46.3% of the unclicked ads. The corresponding frequency for the SGD classifier is 10.8%.

Do note that the relatively high prediction accuracies in Figure 3 are a reflection of sample bias to some extent. Since ads were chosen for display because they were deemed likely to be clicked they are not a completely random test set. Still, the point to be made here is that oLBFGS succeeds in finding an optimal classifier when SGD fails. It would take the processing of about 10^6 feature vectors for SGD to achieve the same accuracy of oLBFGs.

5. Conclusions

An online limited memory version of the (oL)BFGS algorithm was studied for solving strongly convex optimization problems with stochastic objectives. Almost sure convergence was established by bounding the traces and determinants of curvature estimation matrices under the assumption that sample functions have well behaved Hessians. The convergence rate of oLBFGS was further determined to be at least of order $O(1/t)$ in expectation. This rate is customary of stochastic optimization algorithms which are limited by their ability to smooth out the noise in stochastic gradient estimates. A detailed comparison between oLBFGS and SGD for training a logistic regressor in a large scale search engine advertising problem was also presented. The numerical tests show that oLBFGS trains the regressor using less than 1% of the data required by SGD to obtain similar classification accuracy.

Acknowledgments

We acknowledge the support of the National Science Foundation (NSF CAREER CCF-0952867) and the Office of Naval Research (ONR N00014-12-1-0997).

Appendix A. Proof of Proposition 1

We begin by observing that the $\mathbf{p}_u$ sequence in (17) is defined so that we can write $\mathbf{p}_{u+1} = \hat{\mathbf{Z}}_{t-u-1}\mathbf{p}_u$ with $\mathbf{p}_0 = \mathbf{p}$. Indeed, use the explicit expression for $\hat{\mathbf{Z}}_{t-u-1}$ in (12) to write the product $\hat{\mathbf{Z}}_{t-u-1}\mathbf{p}_u$ as

$$\hat{\mathbf{Z}}_{t-u-1}\mathbf{p}_u \;=\; \Big(\mathbf{I} - \hat{\rho}_{t-u-1}\hat{\mathbf{r}}_{t-u-1}\mathbf{v}_{t-u-1}^T\Big)\mathbf{p}_u \;=\; \mathbf{p}_u - \alpha_u\hat{\mathbf{r}}_{t-u-1} \;=\; \mathbf{p}_{u+1}, \tag{42}$$

where the second equality follows from the definition $\alpha_u := \hat{\rho}_{t-u-1}\mathbf{v}_{t-u-1}^T\mathbf{p}_u$ and the third equality from the definition of the $\mathbf{p}_u$ sequence in (17).

Recall now the oLBFGS Hessian inverse approximation expression in (16). It follows that for computing the product $\hat{\mathbf{B}}_t^{-1}\mathbf{p}$ we can multiply each of the $\tau+1$ summands in the right hand side of (16) by $\mathbf{p} = \mathbf{p}_0$. Implementing this procedure yields

$$\begin{aligned}\hat{\mathbf{B}}_t^{-1}\mathbf{p} = \Big(\hat{\mathbf{Z}}_{t-1}^T\ldots\hat{\mathbf{Z}}_{t-\tau}^T\Big)\hat{\mathbf{B}}_{t,0}^{-1}\Big(\hat{\mathbf{Z}}_{t-\tau}\ldots\hat{\mathbf{Z}}_{t-1}\Big)\,\mathbf{p}_0 + \hat{\rho}_{t-\tau}\Big(\hat{\mathbf{Z}}_{t-1}^T\ldots\hat{\mathbf{Z}}_{t-\tau+1}^T\Big)\,\mathbf{v}_{t-\tau}\mathbf{v}_{t-\tau}^T\Big(\hat{\mathbf{Z}}_{t-\tau+1}\ldots\hat{\mathbf{Z}}_{t-1}\Big)\,\mathbf{p}_0\\ + \cdots + \hat{\rho}_{t-2}\Big(\hat{\mathbf{Z}}_{t-1}^T\Big)\,\mathbf{v}_{t-2}\mathbf{v}_{t-2}^T\Big(\hat{\mathbf{Z}}_{t-1}\Big)\,\mathbf{p}_0 + \hat{\rho}_{t-1}\mathbf{v}_{t-1}\mathbf{v}_{t-1}^T\mathbf{p}_0.\end{aligned} \tag{43}$$

The fundamental observation in (43) is that all summands except the last contain the product $\hat{\mathbf{Z}}_{t-1}\mathbf{p}_0$. This product cannot only be computed efficiently but, as shown in (42), is given by $\mathbf{p}_1 = \hat{\mathbf{Z}}_{t-1}\mathbf{p}_0$. A not so fundamental, yet still important observation, is that the last term can be simplified to $\hat{\rho}_{t-1}\mathbf{v}_{t-1}\mathbf{v}_{t-1}^T\mathbf{p}_0 = \alpha_0\mathbf{v}_{t-1}$ given the definition of $\alpha_0 := \hat{\rho}_{t-1}\mathbf{v}_{t-1}^T\mathbf{p}_0$. Implementing both of these substitutions in (43) yields

$$\begin{aligned}\hat{\mathbf{B}}_t^{-1}\mathbf{p} = \Big(\hat{\mathbf{Z}}_{t-1}^T\ldots\hat{\mathbf{Z}}_{t-\tau}^T\Big)\hat{\mathbf{B}}_{t,0}^{-1}\Big(\hat{\mathbf{Z}}_{t-\tau}\ldots\hat{\mathbf{Z}}_{t-2}\Big)\,\mathbf{p}_1 + \hat{\rho}_{t-\tau}\Big(\hat{\mathbf{Z}}_{t-1}^T\ldots\hat{\mathbf{Z}}_{t-\tau+1}^T\Big)\,\mathbf{v}_{t-\tau}\mathbf{v}_{t-\tau}^T\Big(\hat{\mathbf{Z}}_{t-\tau+1}\ldots\hat{\mathbf{Z}}_{t-2}\Big)\,\mathbf{p}_1\\ + \cdots + \hat{\rho}_{t-2}\Big(\hat{\mathbf{Z}}_{t-1}^T\Big)\,\mathbf{v}_{t-2}\mathbf{v}_{t-2}^T\mathbf{p}_1 + \alpha_0\mathbf{v}_{t-1}.\end{aligned} \tag{44}$$

The structure of (44) is analogous to the structure of (43). In all terms except the last two we require determination of the product $\hat{\mathbf{Z}}_{t-2}\mathbf{p}_1$, which, as per (42) can be computed with $2n$ multiplications and is given by $\mathbf{p}_2 = \hat{\mathbf{Z}}_{t-2}\mathbf{p}_1$. Likewise, in the second to last term we can simplify the product $\hat{\rho}_{t-2}\mathbf{v}_{t-2}\mathbf{v}_{t-2}^T\mathbf{p}_1 = \alpha_1\mathbf{v}_{t-2}$ using the definition $\alpha_1 = \hat{\rho}_{t-2}\mathbf{v}_{t-2}^T\mathbf{p}_1$. Implementing these substitutions in (44) yields an expression that is, again, analogous. In all of the resulting summands except the last three we need to compute the product $\hat{\mathbf{Z}}_{t-3}\mathbf{p}_2$, which is given by $\mathbf{p}_3 = \hat{\mathbf{Z}}_{t-3}\mathbf{p}_2$ and in the third to last term we can simplify the product $\hat{\rho}_{t-3}\mathbf{v}_{t-3}\mathbf{v}_{t-3}^T\mathbf{p}_2 = \alpha_2\mathbf{v}_{t-3}$. Repeating this process keeps yielding terms with analogous structure and, after $\tau-1$ repetitions we simplify (44) to

$$\hat{\mathbf{B}}_t^{-1}\mathbf{p} = \Big(\hat{\mathbf{Z}}_{t-1}^T\ldots\hat{\mathbf{Z}}_{t-\tau+1}^T\hat{\mathbf{Z}}_{t-\tau}^T\Big)\,\hat{\mathbf{B}}_{t,0}^{-1}\mathbf{p}_\tau + \Big(\hat{\mathbf{Z}}_{t-1}^T\ldots\hat{\mathbf{Z}}_{t-\tau+1}^T\Big)\,\alpha_{\tau-1}\mathbf{v}_{t-\tau} + \ldots + \hat{\mathbf{Z}}_{t-1}^T\alpha_1\mathbf{v}_{t-2} + \alpha_0\mathbf{v}_{t-1}. \tag{45}$$

In the first summand in (45) we can substitute the definition of the first element of the $\mathbf{q}_u$ sequence $\mathbf{q}_0 := \hat{\mathbf{B}}_{t,0}^{-1}\mathbf{p}_\tau$. More important, observe that the matrix $\hat{\mathbf{Z}}_{t-1}^T$ is the first factor in all but the last summand. Likewise, the matrix $\hat{\mathbf{Z}}_{t-2}^T$ is the second factor in all but the last two summands and, in general, the matrix $\hat{\mathbf{Z}}_{t-u}^T$ is the uth factor in all but the last u summands. Pulling these common factors recursively through (45) it follows that $\hat{\mathbf{B}}_t^{-1}\mathbf{p}_t$ can be equivalently written as

$$\hat{\mathbf{B}}_t^{-1}\mathbf{p} = \alpha_0\mathbf{v}_{t-1} + \hat{\mathbf{Z}}_{t-1}^T\Bigg[\alpha_1\mathbf{v}_{t-2} + \hat{\mathbf{Z}}_{t-2}^T\Bigg[\ldots\Big[\alpha_{\tau-2}\mathbf{v}_{t-\tau+1} + \hat{\mathbf{Z}}_{t-\tau+1}^T\big[\alpha_{\tau-1}\mathbf{v}_{t-\tau} + \hat{\mathbf{Z}}_{t-\tau}^T\mathbf{q}_0\big]\Big]\ldots\Bigg]\Bigg]. \tag{46}$$

To conclude the proof we just need to note that the recursive definition of $\mathbf{q}_u$ in (18) is a computation of the nested elements of (46). To see this consider the innermost element of (46) and use the

definition of $\beta_0 := \hat{\rho}_{t-\tau}\hat{\mathbf{r}}_{t-\tau}^T\mathbf{q}_0$ to conclude that $\alpha_{\tau-1}\mathbf{v}_{t-\tau} + \hat{\mathbf{Z}}_{t-\tau}^T\mathbf{q}_0$ is given by

$$\alpha_{\tau-1}\mathbf{v}_{t-\tau} + \hat{\mathbf{Z}}_{t-\tau}^T\mathbf{q}_0 \;=\; \alpha_{\tau-1}\mathbf{v}_{t-\tau} + \mathbf{q}_0 - \hat{\rho}_{t-\tau}\mathbf{v}_{t-\tau}\hat{\mathbf{r}}_{t-\tau}^T\mathbf{q}_0 \;=\; \mathbf{q}_0 + (\alpha_{\tau-1} - \beta_0)\mathbf{v}_{t-\tau} \;=\; \mathbf{q}_1 \quad (47)$$

where in the last equality we use the definition of $\mathbf{q}_1$ [cf. (18). Substituting this simplification into (46) eliminates the innermost nested term and leads to

$$\hat{\mathbf{B}}_t^{-1}\mathbf{p} = \alpha_0\mathbf{v}_{t-1} + \hat{\mathbf{Z}}_{t-1}^T\Bigg[\alpha_1\mathbf{v}_{t-2} + \hat{\mathbf{Z}}_{t-2}^T\Big[\dots\Big[\alpha_{\tau-2}\mathbf{v}_{t-\tau+1} + \hat{\mathbf{Z}}_{t-\tau+1}^T\mathbf{q}_1\Big]\dots\Big]\Bigg]. \quad (48)$$

Mimicking the computations in (47) we can see that the innermost term in (48) is $\alpha_{\tau-2}\mathbf{v}_{t-\tau+1} + \hat{\mathbf{Z}}_{t-\tau+1}^T\mathbf{q}_1 = \mathbf{q}_2$ and obtain an analogous expression that we can substitute for $\mathbf{q}_3$ and so on. Repeating this process $\tau - 2$ times leads to the last term being $\hat{\mathbf{B}}_t^{-1}\mathbf{p} = \alpha_0\mathbf{v}_{t-1} + \hat{\mathbf{Z}}_{t-1}^T\mathbf{q}_{\tau-1}$ which we can write as $\alpha_0\mathbf{v}_{t-1} + \hat{\mathbf{Z}}_{t-1}^T\mathbf{q}_{\tau-1} = \mathbf{q}_\tau$ by repeating the operations in (47). This final observation yields $\hat{\mathbf{B}}_t^{-1}\mathbf{p} = \mathbf{q}_\tau$.

Appendix B. Proof of Lemma 2

As per (22) in Assumption 1 the eigenvalues of the instantaneous Hessian $\hat{\mathbf{H}}(\mathbf{w}, \tilde{\boldsymbol{\theta}})$ are bounded by $\tilde{m}$ and $\tilde{M}$. Thus, for any given vector $\mathbf{z}$ it holds

$$\tilde{m}\|\mathbf{z}\|^2 \leq \mathbf{z}^T\hat{\mathbf{H}}(\mathbf{w}, \tilde{\boldsymbol{\theta}})\mathbf{z} \leq \tilde{M}\|\mathbf{z}\|^2. \quad (49)$$

For given $\mathbf{w}_t$ and $\mathbf{w}_{t+1}$ define the mean instantaneous Hessian $\hat{\mathbf{G}}_t$ as the average Hessian value along the segment $[\mathbf{w}_t, \mathbf{w}_{t+1}]$

$$\hat{\mathbf{G}}_t = \int_0^1 \hat{\mathbf{H}}\left(\mathbf{w}_t + \tau(\mathbf{w}_{t+1} - \mathbf{w}_t), \tilde{\boldsymbol{\theta}}_t\right) d\tau. \quad (50)$$

Consider now the instantaneous gradient $\hat{\mathbf{s}}(\mathbf{w}_t + \tau(\mathbf{w}_{t+1} - \mathbf{w}_t), \tilde{\boldsymbol{\theta}}_t)$ evaluated at $\mathbf{w}_t + \tau(\mathbf{w}_{t+1} - \mathbf{w}_t)$ and observe that its derivative with respect to τ is $\partial\hat{\mathbf{s}}\big(\mathbf{w}_t + \tau(\mathbf{w}_{t+1} - \mathbf{w}_t), \tilde{\boldsymbol{\theta}}_t\big)/\partial\tau = \hat{\mathbf{H}}(\mathbf{w}_t + \tau(\mathbf{w}_{t+1} - \mathbf{w}_t), \tilde{\boldsymbol{\theta}}_t)(\mathbf{w}_{t+1} - \mathbf{w}_t)$. Then according to the fundamental theorem of calculus

$$\int_0^1 \hat{\mathbf{H}}\left(\mathbf{w}_t + \tau(\mathbf{w}_{t+1} - \mathbf{w}_t)\,,\, \tilde{\boldsymbol{\theta}}_t\right)(\mathbf{w}_{t+1} - \mathbf{w}_t)\; d\tau \;=\; \hat{\mathbf{s}}(\mathbf{w}_{t+1}, \tilde{\boldsymbol{\theta}}_t) - \hat{\mathbf{s}}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t). \quad (51)$$

Using the definitions of the mean instantaneous Hessian $\hat{\mathbf{G}}_t$ in (50) as well as the definitions of the stochastic gradient variations $\hat{\mathbf{r}}_t$ and variable variations $\mathbf{v}_t$ in (11) and (4) we can rewrite (51) as

$$\hat{\mathbf{G}}_t\mathbf{v}_t = \hat{\mathbf{r}}_t. \quad (52)$$

Invoking (49) for the integrand in (50), i.e., for $\hat{\mathbf{H}}(\mathbf{w}, \tilde{\boldsymbol{\theta}}) = \hat{\mathbf{H}}\big(\mathbf{w}_t + \tau(\mathbf{w}_{t+1} - \mathbf{w}_t), \tilde{\boldsymbol{\theta}}\big)$, it follows that for all vectors $\mathbf{z}$ the mean instantaneous Hessian $\hat{\mathbf{G}}_t$ satisfies

$$\tilde{m}\|\mathbf{z}\|^2 \leq \mathbf{z}^T\hat{\mathbf{G}}_t\mathbf{z} \leq \tilde{M}\|\mathbf{z}\|^2. \quad (53)$$

The claim in (26) follows from (52) and (53). Indeed, consider the ratio of inner products $\hat{\mathbf{r}}_t^T\mathbf{v}_t/\mathbf{v}_t^T\mathbf{v}_t$ and use (52) and the first inequality in (53) to write

$$\frac{\hat{\mathbf{r}}_t^T\mathbf{v}_t}{\mathbf{v}_t^T\mathbf{v}_t} = \frac{\mathbf{v}_t^T\hat{\mathbf{G}}_t\mathbf{v}_t}{\mathbf{v}_t^T\mathbf{v}_t} \geq \tilde{m}. \quad (54)$$

It follows that (26) is true for all times t.

To prove (27) we operate (52) and (53). Considering the ratio of inner products $\hat{\mathbf{r}}_t^T\hat{\mathbf{r}}_t/\hat{\mathbf{r}}_t^T\mathbf{v}_t$ and observing that (52) states $\hat{\mathbf{G}}_t\mathbf{v}_t = \hat{\mathbf{r}}_t$, we can write

$$\frac{\hat{\mathbf{r}}_t^T\hat{\mathbf{r}}_t}{\hat{\mathbf{r}}_t^T\mathbf{v}_t} = \frac{\mathbf{v}_t^T\hat{\mathbf{G}}_t^2\mathbf{v}_t}{\mathbf{v}_t^T\hat{\mathbf{G}}_t\mathbf{v}_t} \tag{55}$$

Since the mean instantaneous Hessian $\hat{\mathbf{G}}_t$ is positive definite according to (53), we can define $\mathbf{z}_t = \hat{\mathbf{G}}_t^{1/2}\mathbf{v}_t$. Substituting this observation into (55) we can conclude

$$\frac{\hat{\mathbf{r}}_t^T\hat{\mathbf{r}}_t}{\hat{\mathbf{r}}_t^T\mathbf{v}_t} = \frac{\mathbf{z}_t^T\hat{\mathbf{G}}_t\mathbf{z}_t}{\mathbf{z}_t^T\mathbf{z}_t}. \tag{56}$$

Observing (56) and the inequalities in (53), it follows that (27) is true.

Appendix C. Proof of Lemma 3

We begin with the trace upper bound in (29). Consider the recursive update formula for the Hessian approximation $\hat{\mathbf{B}}_t$ as defined in (28). To simplify notation we define s as a new index such that $s = t - \tau + u$. Introduce this simplified notation in (28) and compute the trace of both sides. Since traces are linear function of their arguments we obtain

$$\text{tr}\left(\hat{\mathbf{B}}_{t,u+1}\right) = \text{tr}\left(\hat{\mathbf{B}}_{t,u}\right) - \text{tr}\left(\frac{\hat{\mathbf{B}}_{t,u}\mathbf{v}_s\mathbf{v}_s^T\hat{\mathbf{B}}_{t,u}}{\mathbf{v}_s^T\hat{\mathbf{B}}_{t,u}\mathbf{v}_s}\right) + \text{tr}\left(\frac{\hat{\mathbf{r}}_s\hat{\mathbf{r}}_s^T}{\mathbf{v}_s^T\hat{\mathbf{r}}_s}\right). \tag{57}$$

Recall that the trace of a matrix product is independent of the order of the factors to conclude that the second summand of (57) can be simplified to

$$\text{tr}\left(\hat{\mathbf{B}}_{t,u}\mathbf{v}_s\mathbf{v}_s^T\hat{\mathbf{B}}_{t,u}\right) \;=\; \text{tr}\left(\mathbf{v}_s^T\hat{\mathbf{B}}_{t,u}\hat{\mathbf{B}}_{t,u}\mathbf{v}_s\right) \;=\; \mathbf{v}_s^T\hat{\mathbf{B}}_{t,u}\hat{\mathbf{B}}_{t,u}\mathbf{v}_s \;=\; \left\|\hat{\mathbf{B}}_{t,u}\mathbf{v}_s\right\|^2, \tag{58}$$

where the second equality follows because $\mathbf{v}_s^T\hat{\mathbf{B}}_{t,u}\hat{\mathbf{B}}_{t,u}\mathbf{v}_s$ is a scalar and the second equality by observing that the term $\mathbf{v}_s^T\hat{\mathbf{B}}_{t,u}\hat{\mathbf{B}}_{t,u}\mathbf{v}_s$ is the inner product of the vector $\hat{\mathbf{B}}_{t,u}\mathbf{v}_s$ with itself. Use the same procedure for the last summand of (57) so as to write $\text{tr}(\hat{\mathbf{r}}_s\hat{\mathbf{r}}_s^T) = \hat{\mathbf{r}}_s^T\hat{\mathbf{r}}_s = \|\hat{\mathbf{r}}_s\|^2$. Substituting this latter observation as well as (58) into (57) we can simplify the trace of $\hat{\mathbf{B}}_{t,u+1}$ to

$$\text{tr}\left(\hat{\mathbf{B}}_{t,u+1}\right) = \text{tr}\left(\hat{\mathbf{B}}_{t,u}\right) - \frac{\|\hat{\mathbf{B}}_{t,u}\mathbf{v}_s\|^2}{\mathbf{v}_s^T\hat{\mathbf{B}}_{t,u}\mathbf{v}_s} + \frac{\|\hat{\mathbf{r}}_s\|^2}{\hat{\mathbf{r}}_s^T\mathbf{v}_s}. \tag{59}$$

The second term in the right hand side of (59) is negative because, as we have already shown, the matrix $\hat{\mathbf{B}}_{t,u}$ is positive definite. The third term is the one for which we have derived the bound that appears in (27) of Lemma 2. Using this two observations we can conclude that the trace of $\hat{\mathbf{B}}_{t,u+1}$ can be bounded as

$$\text{tr}\left(\hat{\mathbf{B}}_{t,u+1}\right) \leq \text{tr}\left(\hat{\mathbf{B}}_{t,u}\right) + \tilde{M}. \tag{60}$$

By considering (60) as a recursive expression for $u = 0, \dots \tau - 1$, we can conclude that

$$\text{tr}\left(\hat{\mathbf{B}}_{t,u}\right) \leq \text{tr}\left(\hat{\mathbf{B}}_{t,0}\right) + u\tilde{M}. \tag{61}$$

To finalize the proof of (29) we need to find a bound for the initial trace $\text{tr}(\hat{\mathbf{B}}_{t,0})$. To do so we consider the definition $\hat{\mathbf{B}}_{t,0} = \mathbf{I}/\hat{\gamma}_t$ with $\hat{\gamma}_t$ as given by (19). Using this definition of $\hat{\mathbf{B}}_{t,0}$ as a scaled identity it follows that we can write the trace of $\hat{\mathbf{B}}_{t,0}$ as

$$\text{tr}\left(\hat{\mathbf{B}}_{t,0}\right) \;=\; \text{tr}\left(\frac{\mathbf{I}}{\hat{\gamma}_t}\right) \;=\; \frac{n}{\hat{\gamma}_t}. \tag{62}$$

Substituting the definition of $\hat{\gamma}_t$ into the rightmost side of (19) it follows that for all times $t \geq 1$,

$$\text{tr}\left(\hat{\mathbf{B}}_{t,0}\right) \;=\; n\,\frac{\hat{\mathbf{r}}_{t-1}^T\hat{\mathbf{r}}_{t-1}}{\mathbf{v}_{t-1}^T\hat{\mathbf{r}}_{t-1}} \;=\; n\,\frac{\|\hat{\mathbf{r}}_{t-1}\|^2}{\mathbf{v}_{t-1}^T\hat{\mathbf{r}}_{t-1}}\,. \tag{63}$$

The term $\|\hat{\mathbf{r}}_{t-1}\|^2/\mathbf{v}_{t-1}^T\hat{\mathbf{r}}_{t-1}$ in (75) is of the same form of the rightmost term in (59). We can then, as we did in going from (59) to (60) apply the bound that we provide in (27) of Lemma 2 to conclude that for all times $t \geq 1$

$$\text{tr}\left(\hat{\mathbf{B}}_{t,0}\right) \;\leq\; n\tilde{M}. \tag{64}$$

Substituting (64) into (61) and pulling common factors leads to the conclusion that for all times $t \geq 1$ and indices $0 \leq u \leq \tau$ it holds

$$\text{tr}\left(\hat{\mathbf{B}}_{t,u}\right) \;\leq\; (n+u)\tilde{M}. \tag{65}$$

The bound in (29) follows by making $u = \tau$ in (65) and recalling that, by definition, $\hat{\mathbf{B}}_t = \hat{\mathbf{B}}_{t,\tau}$. For time $t = 0$ we have $\hat{\gamma}_t = \hat{\gamma}_0 = 1$ and (75) reduces to $\text{tr}(\hat{\mathbf{B}}_{t,0}) = n$ while (65) reduces to $\text{tr}(\hat{\mathbf{B}}_{t,\tau}) \leq (1+\tau)\tilde{M}$. Furthermore, for $t < \tau$ we make $\hat{\mathbf{B}}_t = \hat{\mathbf{B}}_{t,t}$ instead of $\hat{\mathbf{B}}_t = \hat{\mathbf{B}}_{t,\tau}$. In this case the bound in (65) can be tightened to $\text{tr}(\hat{\mathbf{B}}_{t,\tau}) \leq (n+t)\tilde{M}$. Given that we are interested in an asymptotic convergence analysis, these bounds are inconsequential.

We consider now the determinant lower bound in (30). As we did in (57) begin by considering the recursive update in (28) and define s as a new index such that $s = t-\tau+u$ to simplify notation. Compute the determinant of both sides of (28), factorize $\hat{\mathbf{B}}_{t,u}$ on the right hand side, and use the fact that the determinant of a product is the product of the determinants to conclude that

$$\det\left(\hat{\mathbf{B}}_{t,u+1}\right) = \det\left(\hat{\mathbf{B}}_{t,u}\right)\det\left(\mathbf{I} - \frac{\mathbf{v}_s(\hat{\mathbf{B}}_{t,u}\mathbf{v}_s)^T}{\mathbf{v}_s^T\hat{\mathbf{B}}_{t,u}\mathbf{v}_s} + \frac{\hat{\mathbf{B}}_{t,u}^{-1}\hat{\mathbf{r}}_s\hat{\mathbf{r}}_s^T}{\hat{\mathbf{r}}_s^T\mathbf{v}_s}\right). \tag{66}$$

To simplify the right hand side of (66) we should first know that for any vectors $\mathbf{u}_1$, $\mathbf{u}_2$, $\mathbf{u}_3$ and $\mathbf{u}_4$, we can write $\det(\mathbf{I} + \mathbf{u}_1\mathbf{u}_2^T + \mathbf{u}_3\mathbf{u}_4^T) = (1+\mathbf{u}_1^T\mathbf{u}_2)(1+\mathbf{u}_3^T\mathbf{u}_4) - (\mathbf{u}_1^T\mathbf{u}_4)(\mathbf{u}_2^T\mathbf{u}_3)$ – see, e.g., Li and Fukushima (2001), Lemma 3.3). Setting $\mathbf{u}_1 = \mathbf{v}_s$, $\mathbf{u}_2 = \hat{\mathbf{B}}_{t,u}\mathbf{v}_s/\mathbf{v}_s^T\hat{\mathbf{B}}_{t,u}\mathbf{v}_s$, $\mathbf{u}_3 = \hat{\mathbf{B}}_{t,u}^{-1}\hat{\mathbf{r}}_s$ and $\mathbf{u}_4 = \hat{\mathbf{r}}_s/\hat{\mathbf{r}}_s^T\mathbf{v}_s$, implies that $\det(\mathbf{I} + \mathbf{u}_1\mathbf{u}_2^T + \mathbf{u}_3\mathbf{u}_4^T)$ is equivalent to the last term in the right hand side of (66). Applying these substitutions implies that $(1+\mathbf{u}_1^T\mathbf{u}_2) = 1 - \mathbf{v}_s^T\hat{\mathbf{B}}_{t,u}\mathbf{v}_s/\mathbf{v}_s\hat{\mathbf{B}}_{t,u}\mathbf{v}_s = 0$ and $\mathbf{u}_1^T\mathbf{u}_4 = -\mathbf{v}_s^T\hat{\mathbf{r}}_s/\hat{\mathbf{r}}_s^T\mathbf{v}_s = -1$. Hence, the term $\det(\mathbf{I} + \mathbf{u}_1\mathbf{u}_2^T + \mathbf{u}_3\mathbf{u}_4^T)$ can be simplified as $\mathbf{u}_2^T\mathbf{u}_3$. By this simplification we can write the right hand side of (66) as

$$\det\left[\mathbf{I} - \frac{\mathbf{v}_s(\hat{\mathbf{B}}_{t,u}\mathbf{v}_s)^T}{\mathbf{v}_s^T\hat{\mathbf{B}}_{t,u}\mathbf{v}_s} + \frac{\hat{\mathbf{B}}_{t,u}^{-1}\hat{\mathbf{r}}_s\hat{\mathbf{r}}_s^T}{\hat{\mathbf{r}}_s^T\mathbf{v}_s}\right] = \frac{\left(\hat{\mathbf{B}}_{t,u}\mathbf{v}_s\right)^T}{\mathbf{v}_s^T\hat{\mathbf{B}}_{t,u}\mathbf{v}_s}\hat{\mathbf{B}}_{t,u}^{-1}\hat{\mathbf{r}}_s. \tag{67}$$

To further simplify (67) write $(\hat{\mathbf{B}}_{t,u}\mathbf{v}_s)^T = \mathbf{v}_s^T\hat{\mathbf{B}}_{t,u}^T$ and observer that since $\hat{\mathbf{B}}_{t,u}$ is symmetric we have $\hat{\mathbf{B}}_{t,u}^T\hat{\mathbf{B}}_{t,u}^{-1} = \hat{\mathbf{B}}_{t,u}\hat{\mathbf{B}}_{t,u}^{-1} = \mathbf{I}$. Therefore,

$$\det\left[\mathbf{I} - \frac{\mathbf{v}_s(\hat{\mathbf{B}}_{t,u}\mathbf{v}_s)^T}{\mathbf{v}_i^T\hat{\mathbf{B}}_{t,u}\mathbf{v}_s} + \frac{\hat{\mathbf{B}}_{t,u}^{-1}\hat{\mathbf{r}}_s\hat{\mathbf{r}}_s^T}{\hat{\mathbf{r}}_s^T\mathbf{v}_s}\right] = \frac{\hat{\mathbf{r}}_s^T\mathbf{v}_s}{\mathbf{v}_s^T\hat{\mathbf{B}}_{t,u}\mathbf{v}_s}. \tag{68}$$

Substitute the simplification in (68) for the corresponding factor in (66). Further multiply and divide the right hand side by the nonzero norm $\|\mathbf{v}_s\|$ and regroup terms to obtain

$$\det\left(\hat{\mathbf{B}}_{t,u+1}\right) = \det\left(\hat{\mathbf{B}}_{t,u}\right)\frac{\hat{\mathbf{r}}_s^T\mathbf{v}_s}{\|\mathbf{v}_s\|}\frac{\|\mathbf{v}_s\|}{\mathbf{v}_s^T\hat{\mathbf{B}}_{t,u}\mathbf{v}_s}. \tag{69}$$

To bound the third factor in (69) observe that the largest possible value for the normalized quadratic form $\mathbf{v}_s^T \hat{\mathbf{B}}_{t,u} \mathbf{v}_s / \|\mathbf{v}_s\|^2$ occurs when $\mathbf{v}_s$ is an eigenvector of $\hat{\mathbf{B}}_{t,u}$ associated with its largest eigenvalue. In such case the value attained is precisely the largest eigenvalue of $\hat{\mathbf{B}}_{t,u}$ implying that we can write

$$\frac{\mathbf{v}_s^T \hat{\mathbf{B}}_{t,u} \mathbf{v}_s}{\|\mathbf{v}_s\|} \leq \lambda_{\max}\left(\hat{\mathbf{B}}_{t,u}\right). \tag{70}$$

But to bound the largest eigenvalue $\lambda_{\max}(\hat{\mathbf{B}}_{t,u})$ we can just use the fact that the trace of a matrix coincides with the sum of its eigenvalues. In particular, it must be that $\lambda_{\max}(\hat{\mathbf{B}}_{t,u}) \leq \text{tr}(\hat{\mathbf{B}}_{t,u})$ because all the eigenvalues of the positive definite matrix $\hat{\mathbf{B}}_{t,u}$ are positive. Combining this observation with the trace bound in (65) leads to

$$\frac{\mathbf{v}_s^T \hat{\mathbf{B}}_{t,u} \mathbf{v}_s}{\|\mathbf{v}_s\|} \;\leq\; \text{tr}\left(\hat{\mathbf{B}}_{t,u}\right) \;\leq\; (n+u)\tilde{M}. \tag{71}$$

We can also bound the second factor in the right hand side of (69) if we reorder the inequality in (26) of Lemma 2 to conclude that $\hat{\mathbf{r}}_s^T \mathbf{v}_s / \|\mathbf{v}_s\| \leq \tilde{m}$. This bound, along with the inverse of the inequality in (71) substituted in (69) leads to

$$\det\left(\hat{\mathbf{B}}_{t,u+1}\right) \geq \frac{\tilde{m}}{n\tilde{M} + u\tilde{M}} \det\left(\hat{\mathbf{B}}_{t,u}\right). \tag{72}$$

Apply (72) recursively between indexes $u = 0$ and $u = \tau - 1$ and further observing that $u \leq \tau$ in all of the resulting factors it follows that

$$\det\left(\hat{\mathbf{B}}_{t,\tau}\right) \;\geq\; \left[\frac{\tilde{m}}{(n+\tau)\tilde{M}}\right]^{\tau} \det\left(\hat{\mathbf{B}}_{t,0}\right). \tag{73}$$

To finalize the derivation of (30) we just need to bound the determinant of the initial curvature approximation matrix $\hat{\mathbf{B}}_{t,0}$. To do so we consider, again, the definition $\hat{\mathbf{B}}_{t,0} = \mathbf{I}/\hat{\gamma}_t$ with $\hat{\gamma}_t$ as given by (19). Using this definition of $\hat{\mathbf{B}}_{t,0}$ as a scaled identity it follows that we can write the determinant of $\hat{\mathbf{B}}_{t,0}$ as

$$\det\left(\hat{\mathbf{B}}_{t,0}\right) \;=\; \det\left(\frac{\mathbf{I}}{\hat{\gamma}_t}\right) \;=\; \frac{1}{\hat{\gamma}_t^n}. \tag{74}$$

Substituting the definition of $\hat{\gamma}_t$ into the rightmost side of (74) it follows that for all times $t \geq 1$,

$$\det\left(\hat{\mathbf{B}}_{t,0}\right) \;=\; \left(\frac{\hat{\mathbf{r}}_{t-1}^T \hat{\mathbf{r}}_{t-1}}{\mathbf{v}_{t-1}^T \hat{\mathbf{r}}_{t-1}}\right)^n \;=\; \left(\frac{\|\hat{\mathbf{r}}_{t-1}\|^2}{\mathbf{v}_{t-1}^T \hat{\mathbf{r}}_{t-1}}\right)^n. \tag{75}$$

The term $\|\hat{\mathbf{r}}_{t-1}\|^2 / \mathbf{v}_{t-1}^T \hat{\mathbf{r}}_{t-1}$ has lower and upper bounds that we provide in (27) of Lemma 2. Using the lower bound in (27) it follows that the initial determinant must be such that

$$\det\left(\hat{\mathbf{B}}_{t,0}\right) \geq \tilde{m}^n. \tag{76}$$

Substituting the upper bound in (76) for the determinant of the initial curvature approximation matrix in (73) allows us to conclude that for all times $t \geq 1$

$$\det\left(\hat{\mathbf{B}}_{t,\tau}\right) \;\geq\; \tilde{m}^n \left[\frac{\tilde{m}}{(n+\tau)\tilde{M}}\right]^{\tau}. \tag{77}$$

The bound in (30) follows by making $u = \tau$ in (77) and recalling that, by definition, $\hat{\mathbf{B}}_t = \hat{\mathbf{B}}_{t,\tau}$. At time $t = 0$ the initialization constant is set to $\hat{\gamma}_t = \hat{\gamma}_0 = 1$ and (76) reduces to $\det(\hat{\mathbf{B}}_{t,0}) = 1$ while (77) reduces to $\det(\hat{\mathbf{B}}_{t,\tau}) \leq [\tilde{m}/(1+\tau)\tilde{M}]^{\tau}$. For $t < \tau$ we make $\hat{\mathbf{B}}_t = \hat{\mathbf{B}}_{t,t}$ instead of $\hat{\mathbf{B}}_t = \hat{\mathbf{B}}_{t,\tau}$. In this case the bound in (65) can be tightened to $\det(\hat{\mathbf{B}}_{t,\tau}) \leq \tilde{m}[\tilde{m}^n/(1+\tau)\tilde{M}]^{\tau}$. As in the case of the trace, given that we are interested in an asymptotic convergence analysis, these bounds are inconsequential.

Appendix D. Proof of Lemma 4

We first prove the upper bound inequality in (31). Let us define λ_i as the ith largest eigenvalue of matrix $\hat{\mathbf{B}}_t$. Considering the result in Lemma 3 that $\text{tr}(\hat{\mathbf{B}}_t) \leq (n+\tau)\tilde{M}$ for all steps $t \geq 1$, we obtain that the sum of eigenvalues of the Hessian approximation $\hat{\mathbf{B}}_t$ satisfy

$$\sum_{i=1}^{n} \lambda_i \;=\; \text{tr}\left(\hat{\mathbf{B}}_t\right) \;\leq\; (n+\tau)\tilde{M}. \tag{78}$$

Considering the upper bound for the sum of eigenvalues in (78) and recalling that all the eigenvalues of the matrix $\hat{\mathbf{B}}_t$ are positive because $\hat{\mathbf{B}}_t$ is positive definite, we can conclude that each of the eigenvalues of $\hat{\mathbf{B}}_t$ is less than the upper bound for their sum in (78). We then have $\lambda_i \leq (n+\tau)\tilde{M}$ for all i from where the right inequality in (31) follows.

To prove the lower bound inequality in (31) consider the second result of Lemma 3 which provides a lower bound for the determinant of the Hessian approximation matrix $\hat{\mathbf{B}}_t$. According to the fact that determinant of a matrix is the product of its eigenvalues, it follows that the product of the eigenvalues of $\hat{\mathbf{B}}_t$ is bounded below by the lower bound in (30), or, equivalently, $\prod_{i=1}^{n} \lambda_i \geq \tilde{m}^{n+\tau}/[(n+\tau)\tilde{M}]^{\tau}$. Hence, for any given eigenvalue of $\hat{\mathbf{B}}_t$, say λ_j, we have

$$\lambda_j \;\geq\; \frac{1}{\prod_{k=1,k\neq j}^{n} \lambda_k} \times \frac{\tilde{m}^{n+\tau}}{\left[(n+\tau)\tilde{M}\right]^{\tau}}. \tag{79}$$

But in the first part of this proof we have already showed that $(n+\tau)\tilde{M}$ is a lower bound for the eigenvalues of $\hat{\mathbf{B}}_t$. We can then conclude that the product of the $n-1$ eigenvalues $\prod_{k=1,k\neq j}^{n} \lambda_k$ is bounded above by $[(n+\tau)\tilde{M}]^{n-1}$, i.e.,

$$\prod_{k=1,k\neq j}^{n} \lambda_k \leq \left[(n+\tau)\tilde{M}\right]^{n-1}. \tag{80}$$

Combining the inequalities in (79) and (80) we conclude that for any specific eigenvalue of $\hat{\mathbf{B}}_t$ can be lower bounded as

$$\lambda_j \;\geq\; \frac{1}{\left[(n+\tau)\tilde{M}\right]^{n-1}} \times \frac{\tilde{m}^{n+\tau}}{\left[(n+\tau)\tilde{M}\right]^{\tau}}. \tag{81}$$

Since inequality (81) is true for all the eigenvalues of $\hat{\mathbf{B}}_t$, the left inequality (31) holds true.

Appendix E. Proof of Lemma 5

The proof is standard in stochastic optimization and provided here for reference. As it follows from Assumption 1 the eigenvalues of the Hessian $\mathbf{H}(\mathbf{w}_t) = \mathbb{E}_{\tilde{\boldsymbol{\theta}}}[\hat{\mathbf{H}}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t)] = \nabla^2_{\mathbf{w}} F(\mathbf{w}_t)$ are bounded between $0 < m$ and $M < \infty$ as stated in (25). Taking a Taylor's expansion of the function $F(\mathbf{w})$ around $\mathbf{w} = \mathbf{w}_t$ and using the upper bound in the Hessian eigenvalues we can write

$$F(\mathbf{w}_{t+1}) \;\leq F(\mathbf{w}_t) + \nabla F(\mathbf{w}_t)^T(\mathbf{w}_{t+1} - \mathbf{w}_t) + \frac{M}{2}\|\mathbf{w}_{t+1} - \mathbf{w}_t\|^2. \tag{82}$$

From the definition of the oLBFGS update in (3) we can write the difference of two consecutive variables $\mathbf{w}_{t+1} - \mathbf{w}_t$ as $-\epsilon_t \hat{\mathbf{B}}_t^{-1} \hat{\mathbf{s}}(\mathbf{w}_t, \tilde{\boldsymbol{\theta}}_t)$. Making this substitution in (82), taking expectation with $\mathbf{w}_t$ given in both sides of the resulting inequality, and observing the fact that when $\mathbf{w}_t$ is given the

Hessian approximation $\hat{\mathbf{B}}_t^{-1}$ is deterministic we can write

$$\mathbb{E}\left[F(\mathbf{w}_{t+1})\,\middle|\,\mathbf{w}_t\right] \leq\ F(\mathbf{w}_t) - \epsilon_t \nabla F(\mathbf{w}_t)^T \hat{\mathbf{B}}_t^{-1} \mathbb{E}\left[\hat{\mathbf{s}}(\mathbf{w}_t,\tilde{\boldsymbol{\theta}}_t)\,\middle|\,\mathbf{w}_t\right] + \frac{\epsilon^2 M}{2}\,\mathbb{E}\left[\left\|\hat{\mathbf{B}}_t^{-1}\hat{\mathbf{s}}(\mathbf{w}_t,\tilde{\boldsymbol{\theta}}_t)\right\|^2\,\middle|\,\mathbf{w}_t\right]. \tag{83}$$

We proceed to bound the third term in the right hand side of (83). Start by observing that the 2-norm of a product is not larger than the product of the 2-norms and that, as noted above, with $\mathbf{w}_t$ given the matrix $\hat{\mathbf{B}}_t^{-1}$ is also given to write

$$\mathbb{E}\left[\left\|\hat{\mathbf{B}}_t^{-1}\hat{\mathbf{s}}(\mathbf{w}_t,\tilde{\boldsymbol{\theta}}_t)\right\|^2\,\middle|\,\mathbf{w}_t\right] \leq\ \left\|\hat{\mathbf{B}}_t^{-1}\right\|^2\ \mathbb{E}\left[\left\|\hat{\mathbf{s}}(\mathbf{w}_t,\tilde{\boldsymbol{\theta}}_t)\right\|^2\middle|\mathbf{w}_t\right]. \tag{84}$$

Notice that, as stated in (32), $1/c$ is an upper bound for the eigenvalues of $\hat{\mathbf{B}}_t^{-1}$. Further observe that the second moment of the norm of the stochastic gradient is bounded by $\mathbb{E}\left[\|\hat{\mathbf{s}}(\mathbf{w}_t,\tilde{\boldsymbol{\theta}}_t)\|^2\,\middle|\,\mathbf{w}_t\right] \leq S^2$, as stated in Assumption 2. These two upper bounds substituted in (84) yield

$$\mathbb{E}\left[\left\|\hat{\mathbf{B}}_t^{-1}\hat{\mathbf{s}}(\mathbf{w}_t,\tilde{\boldsymbol{\theta}}_t)\right\|^2\,\middle|\,\mathbf{w}_t\right] \leq \frac{S^2}{c^2}. \tag{85}$$

Substituting the upper bound in (85) for the third term of (83) and further using the fact that $\mathbb{E}\left[\hat{\mathbf{s}}(\mathbf{w}_t,\tilde{\boldsymbol{\theta}}_t)\,\middle|\,\mathbf{w}_t\right] = \nabla F(\mathbf{w}_t)$ in the second term leads to

$$\mathbb{E}\left[F(\mathbf{w}_{t+1})\,\middle|\,\mathbf{w}_t\right] \leq F(\mathbf{w}_t) - \epsilon_t \nabla F(\mathbf{w}_t)^T \hat{\mathbf{B}}_t^{-1} \nabla F(\mathbf{w}_t) + \frac{\epsilon_t^2 M S^2}{2c^2}. \tag{86}$$

We now find a lower bound for the second term in the right hand side of (86). As stated in (32), $1/C$ is a lower bound for the eigenvalues of $\hat{\mathbf{B}}_t^{-1}$. This lower bound implies that

$$\nabla F(\mathbf{w}_t)^T \hat{\mathbf{B}}_t^{-1} \nabla F(\mathbf{w}_t) \geq \frac{1}{C}\|\nabla F(\mathbf{w}_t)\|^2. \tag{87}$$

By substituting the lower bound in (87) for the corresponding summand in (86) we obtain

$$\mathbb{E}\left[F(\mathbf{w}_{t+1})\,\middle|\,\mathbf{w}_t\right] \leq F(\mathbf{w}_t) - \frac{\epsilon_t}{C}\|\nabla F(\mathbf{w}_t)\|^2 + \frac{M S^2 \epsilon_t^2}{2c^2}. \tag{88}$$

Subtracting the optimal objective function value $F(\mathbf{w}^*)$ from the both sides of (88) follows (33).

Appendix F. Proof of Theorem 6

The proof uses the relationship in the statement (33) of Lemma 5 to build a supermartingale sequence. This is also a standard technique in stochastic optimization and provided here for reference. To construct the supermartingale sequence define the stochastic process α_t with values

$$\alpha_t := F(\mathbf{w}_t) - F(\mathbf{w}^*) + \frac{MS^2}{2c^2}\sum_{u=t}^{\infty}\epsilon_u^2. \tag{89}$$

Observe that α_t is well defined because the $\sum_{u=t}^{\infty}\epsilon_u^2 < \sum_{u=0}^{\infty}\epsilon_u^2 < \infty$ is summable. Further define the sequence β_t with values

$$\beta_t :=\ \frac{\epsilon_t}{C}\,\|\nabla F(\mathbf{w}_t)\|^2. \tag{90}$$

Let now $\mathcal{F}_t$ be a sigma-algebra measuring α_t, β_t, and $\mathbf{w}_t$. The conditional expectation of α_{t+1} given $\mathcal{F}_t$ can be written as

$$\mathbb{E}\left[\alpha_{t+1}\,\middle|\,\mathcal{F}_t\right] = \mathbb{E}\left[F(\mathbf{w}_{t+1})\,\middle|\,\mathcal{F}_t\right] - F(\mathbf{w}^*) + \frac{MS^2}{2c^2}\sum_{u=t+1}^{\infty}\epsilon_u^2, \tag{91}$$

because the term $(MS^2/2c^2)\sum_{u=t+1}^{\infty}\epsilon_u^2$ is just a deterministic constant. Substituting (88) of Lemma 5 into (91) and using the definitions of α_t in (89) and β_t in (90) yields

$$\mathbb{E}\left[\alpha_{t+1}\,\middle|\,\alpha_t\right] \;\leq\; \alpha_t - \beta_t. \tag{92}$$

Since the sequences α_t and β_t are nonnegative it follows from (92) that they satisfy the conditions of the supermartingale convergence theorem – see e.g. (Theorem E7.4 in Solo and Kong (1995)) . Therefore, we conclude that: (i) The sequence α_t converges almost surely. (ii) The sum $\sum_{t=0}^{\infty}\beta_t < \infty$ is almost surely finite. Using the explicit form of β_t in (90) we have that $\sum_{t=0}^{\infty}\beta_t < \infty$ is equivalent to

$$\sum_{t=0}^{\infty}\frac{\epsilon_t}{C}\,\|\nabla F(\mathbf{w}_t)\|^2 < \infty, \qquad \text{a.s.} \tag{93}$$

Since the sequence of step sizes is nonsummable, for (93) to be true we need to have a vanishing subsequence embedded in $\|\nabla F(\mathbf{w}_t)\|^2$. By definition, this implies that the limit infimum of the sequence $\|\nabla F(\mathbf{w}_t)\|^2$ is null almost surely,

$$\liminf_{t\to\infty}\|\nabla F(\mathbf{w}_t)\|^2 = 0, \qquad \text{a.s.} \tag{94}$$

We transform the gradient bound in (94) into a bound pertaining to the objective function value optimality $F(\mathbf{w}_t) - F(\mathbf{w}^*)$. To do so, simply observe that the strong convexity of the average function F implies that for any points $\mathbf{z}$ and $\mathbf{y}$

$$F(\mathbf{y}) \geq\; F(\mathbf{z}) + \nabla F(\mathbf{z})^T(\mathbf{y}-\mathbf{z}) + \frac{m}{2}\|\mathbf{y}-\mathbf{z}\|^2. \tag{95}$$

For fixed $\mathbf{z}$, the right hand side of (95) is a quadratic function of $\mathbf{y}$ whose minimum argument we can find by setting its gradient to zero. Doing this yields the minimizing argument $\hat{\mathbf{y}} = \mathbf{z} - (1/m)\nabla F(\mathbf{z})$ implying that for all $\mathbf{y}$ we must have

$$\begin{aligned} F(\mathbf{y}) &\geq\; F(\mathbf{z}) + \nabla F(\mathbf{z})^T(\hat{\mathbf{y}}-\mathbf{z}) + \frac{m}{2}\|\hat{\mathbf{y}}-\mathbf{z}\|^2 \\ &=\; F(\mathbf{z}) - \frac{1}{2m}\|\nabla F(\mathbf{z})\|^2. \end{aligned} \tag{96}$$

Observe that the bound in (96) holds true for all $\mathbf{y}$ and $\mathbf{z}$. Setting values $\mathbf{y} = \mathbf{w}^*$ and $\mathbf{z} = \mathbf{w}_t$ in (96) and rearranging the terms yields a lower bound for the squared gradient norm $\|\nabla F(\mathbf{x}^t)\|^2$ as

$$\|\nabla F(\mathbf{w}_t)\|^2 \geq 2m(F(\mathbf{w}_t) - F(\mathbf{w}^*)). \tag{97}$$

Notice that according to the result in (94) a subsequence of $\|\nabla F(\mathbf{w}_t)\|^2$ converges to null and $\liminf_{t\to\infty}\|\nabla F(\mathbf{w}_t)\|^2 = 0$ almost surely. Observing the relationship in (97), we can conclude that a subsequence of the objective value error $F(\mathbf{w}_t) - F(\mathbf{w}^*)$ sequence converges to null which implies

$$\liminf_{t\to\infty}\; F(\mathbf{w}_t) - F(\mathbf{w}^*) = 0, \qquad \text{a.s.} \tag{98}$$

Based on the martingale convergence theorem for the sequences α_t and β_t in relation (92), the sequence α_t almost surely converges to a limit. Consider the definition of α_t in (89) and observe that the sum $\sum_{u=t}^{\infty}(\gamma^u)^2$ is deterministic and its limit is null. Therefore, the limit $\lim_{t\to\infty} F(\mathbf{w}_t) - F(\mathbf{w}^*)$ of the nonnegative objective function errors $F(\mathbf{w}_t) - F(\mathbf{w}^*)$ almost surely exists. This observation in association with the result in (99) implies that the whole sequence of $F(\mathbf{w}_t) - F(\mathbf{w}^*)$ converges almost surely to zero,

$$\lim_{t\to\infty}\; F(\mathbf{w}_t) - F(\mathbf{w}^*) = 0, \qquad \text{a.s.} \tag{99}$$

The result in (99) holds because the sequence $F(\mathbf{w}_t) - F(\mathbf{w}^*)$ converges almost surely to a limit, while a subsequence of this sequence converges to zero with probability 1 as stated in (98). Combining

these two observations, the limit that the whole sequence converges to should be 0. To transform the objective function optimality bound in (99) into a bound pertaining to the squared distance to optimality $\|\mathbf{w}_t - \mathbf{w}^*\|^2$ simply observe that the lower bound m on the eigenvalues of $\mathbf{H}(\mathbf{w}^*)$ applied to a Taylor's expansion around the optimal argument $\mathbf{w}_t$ implies that

$$F(\mathbf{w}_t) \geq F(\mathbf{w}^*) + \nabla F(\mathbf{w}^*)^T(\mathbf{w}_t - \mathbf{w}^*) + \frac{m}{2}\|\mathbf{w}_t - \mathbf{w}^*\|^2. \tag{100}$$

Notice that the optimal point gradient $\nabla F(\mathbf{x}^*)$ is null. This observation and rearranging the terms in (100) imply that

$$F(\mathbf{w}_t) - F(\mathbf{w}^*) \geq \frac{m}{2}\|\mathbf{w}_t - \mathbf{w}^*\|^2. \tag{101}$$

The upper bound in (101) for the squared norm $\|\mathbf{w}_t - \mathbf{w}^*\|^2$ in association with the fact that the sequence $F(\mathbf{w}_t) - F(\mathbf{w}^*)$ almost surely converges to null, leads to the conclusion that the sequence $\|\mathbf{w}_t - \mathbf{w}^*\|^2$ almost surely converges to null. Hence, the claim in (34) is valid.

Appendix G. Proof of Theorem 7

The proof follows along the lines of (Mokhtari and Ribeiro (2014a)) and is presented here for completeness. Theorem 7 claims that the sequence of expected objective values $\mathbb{E}[F(\mathbf{w}_t)]$ approaches the optimal objective $F(\mathbf{w}^*)$ at a sublinear rate $O(1/t)$. Before proceeding to the proof of Theorem 7 we repeat a technical lemma of (Mokhtari and Ribeiro (2014a)) that provides a sufficient condition for a sequence u_t to exhibit a sublinear convergence rate.

Lemma 8 (Mokhtari and Ribeiro (2014a)) *Let $a > 1$, $b > 0$ and $t_0 > 0$ be given constants and $u_t \geq 0$ be a nonnegative sequence that satisfies the inequality*

$$u_{t+1} \leq \left(1 - \frac{a}{t + t_0}\right) u_t + \frac{b}{(t + t_0)^2}\ , \tag{102}$$

for all times $t \geq 0$. The sequence u_t is then bounded as

$$u_t \leq\ \frac{Q}{t + t_0}, \tag{103}$$

for all times $t \geq 0$, where the constant Q is defined as

$$Q := \max\left[\frac{b}{a-1},\ t_0 u_0\right]. \tag{104}$$

Proof We prove (103) using induction. To prove the claim for $t = 0$ simply observe that the definition of Q in (104) implies that

$$Q := \max\left[\frac{b}{a-1},\ t_0 u_0\right] \geq\ t_0 u_0, \tag{105}$$

because the maximum of two numbers is at least equal to both of them. By rearranging the terms in (105) we can conclude that

$$u_0\ \leq\ \frac{Q}{t_0}. \tag{106}$$

Comparing (106) and (103) it follows that the latter inequality is true for $t = 0$.

Introduce now the induction hypothesis that (103) is true for $t = s$. To show that this implies that (103) is also true for $t = s + 1$ substitute the induction hypothesis $u_s \leq Q/(s + t_0)$ into the recursive relationship in (102). This substitution shows that u_{s+1} is bounded as

$$u_{s+1} \leq \left(1 - \frac{a}{s + t_0}\right) \frac{Q}{s + t_0} + \frac{b}{(s + t_0)^2}\ . \tag{107}$$

Observe now that according to the definition of Q in (104), we know that $b/(a-1) \leq Q$ because Q is the maximum of $b/(a-1)$ and $t_0 u_0$. Reorder this bound to show that $b \leq Q(a-1)$ and substitute into (107) to write

$$u_{s+1} \leq \left(1 - \frac{a}{s+t_0}\right) \frac{Q}{s+t_0} + \frac{(a-1)Q}{(s+t_0)^2} \ . \tag{108}$$

Pulling out $Q/(s+t_0)^2$ as a common factor and simplifying and reordering terms it follows that (108) is equivalent to

$$u_{s+1} \ \leq \ \frac{Q\big[s+t_0-a+(a-1)\big]}{(s+t_0)^2} \ = \ \frac{s+t_0-1}{(s+t_0)^2}\, Q. \tag{109}$$

To complete the induction step use the difference of squares formula for $(s+t_0)^2 - 1$ to conclude that

$$\big[(s+t_0)-1\big]\big[(s+t_0)+1\big] \ = \ (s+t_0)^2 - 1 \ \leq \ (s+t_0)^2. \tag{110}$$

Reordering terms in (110) it follows that $\big[(s+t_0)-1\big]/(s+t_0)^2 \leq 1/\big[(s+t_0)+1\big]$, which upon substitution into (109) leads to the conclusion that

$$u_{s+1} \leq \frac{Q}{s+t_0+1}. \tag{111}$$

Eq. (111) implies that the assumed validity of (103) for $t = s$ implies the validity of (103) for $t = s+1$. Combined with the validity of (103) for $t = 0$, which was already proved, it follows that (103) is true for all times $t \geq 0$. ■

Lemma 8 shows that satisfying (102) is sufficient for a sequence to have the sublinear rate of convergence specified in (103). In the following proof of Theorem 7 we show that if the step size sequence parameters ϵ_0 and T_0 satisfy $2\epsilon_0 T_0/C > 1$ the sequence $\mathbb{E}\left[F(\mathbf{w}_t)\right] - F(\mathbf{w}^*)$ of expected optimality gaps satisfies (102) with $a = 2\epsilon_0 T_0/C$, $b = \epsilon_0^2 T_0^2 M S^2/2c^2$ and $t_0 = T_0$. The result in (35) then follows as a direct consequence of Lemma 8.

Proof of Theorem 7: Consider the result in (88) of Lemma 5 and subtract the average function optimal value $F(\mathbf{w}^*)$ from both sides of the inequality to conclude that the sequence of optimality gaps in the RES algorithm satisfies

$$\mathbb{E}\left[F(\mathbf{w}_{t+1}) \,\middle|\, \mathbf{w}_t\right] - \ F(\mathbf{w}^*) \ \leq \ F(\mathbf{w}_t) - \ F(\mathbf{w}^*) - \frac{\epsilon_t}{C}\|\nabla F(\mathbf{w}_t)\|^2 + \frac{\epsilon_t^2 M S^2}{2c^2}. \tag{112}$$

We proceed to find a lower bound for the gradient norm $\|\nabla F(\mathbf{w}_t)\|$ in terms of the error of the objective value $F(\mathbf{w}_t) - \ F(\mathbf{w}^*)$ – this is a standard derivation which we include for completeness, see, e.g., Boyd and Vandenberghe (2004). As it follows from Assumption 1 the eigenvalues of the Hessian $\mathbf{H}(\mathbf{w}_t)$ are bounded between $0 < m$ and $M < \infty$ as stated in (25). Taking a Taylor's expansion of the objective function $F(\mathbf{y})$ around $\mathbf{w}$ and using the lower bound in the Hessian eigenvalues we can write

$$F(\mathbf{y}) \ \geq \ F(\mathbf{w}) + \nabla F(\mathbf{w})^T(\mathbf{y}-\mathbf{w}) + \frac{m}{2}\|\mathbf{y}-\mathbf{w}\|^2. \tag{113}$$

For fixed $\mathbf{w}$, the right hand side of (113) is a quadratic function of $\mathbf{y}$ whose minimum argument we can find by setting its gradient to zero. Doing this yields the minimizing argument $\hat{\mathbf{y}} = \mathbf{w} - (1/m)\nabla F(\mathbf{w})$ implying that for all $\mathbf{y}$ we must have

$$\begin{aligned} F(\mathbf{y}) \ &\geq \ F(\mathbf{w}) + \nabla F(\mathbf{w})^T(\hat{\mathbf{y}}-\mathbf{w}) + \frac{m}{2}\|\hat{\mathbf{y}}-\mathbf{w}\|^2 \\ &= \ F(\mathbf{w}) - \frac{1}{2m}\|\nabla F(\mathbf{w})\|^2. \end{aligned} \tag{114}$$

The bound in (114) is true for all $\mathbf{w}$ and $\mathbf{y}$. In particular, for $\mathbf{y} = \mathbf{w}^*$ and $\mathbf{w} = \mathbf{w}_t$ (114) yields

$$F(\mathbf{w}^*) \;\geq\; F(\mathbf{w}_t) - \frac{1}{2m}\|\nabla F(\mathbf{w}_t)\|^2. \tag{115}$$

Rearrange terms in (115) to obtain a bound on the gradient norm squared $\|\nabla F(\mathbf{w}_t)\|^2$. Further substitute the result in (112) and regroup terms to obtain the bound

$$\mathbb{E}\left[F(\mathbf{w}_{t+1}) \,\middle|\, \mathbf{w}_t\right] - \; F(\mathbf{w}^*) \;\leq\; \left(1 - \frac{2m\epsilon_t}{C}\right)\big(F(\mathbf{w}_t) - \; F(\mathbf{w}^*)\big) + \frac{\epsilon_t^2 M S^2}{2c^2}. \tag{116}$$

Take now expected values on both sides of (116). The resulting double expectation in the left hand side simplifies to $\mathbb{E}\left[\mathbb{E}\left[F(\mathbf{w}_{t+1}) \,\middle|\, \mathbf{w}_t\right]\right] = \mathbb{E}\left[F(\mathbf{w}_{t+1})\right]$, which allow us to conclude that (116) implies that

$$\mathbb{E}\left[F(\mathbf{w}_{t+1})\right] - \; F(\mathbf{w}^*) \;\leq\; \left(1 - \frac{2m\epsilon_t}{C}\right)\big(\mathbb{E}\left[F(\mathbf{w}_t)\right] - \; F(\mathbf{w}^*)\big) + \frac{\epsilon_t^2 M S^2}{2c^2}. \tag{117}$$

Further substituting $\epsilon_t = \epsilon_0 T_0/(T_0 + t)$, which is the assumed form of the step size sequence by hypothesis, we can rewrite (117) as

$$\mathbb{E}\left[F(\mathbf{w}_{t+1})\right] - \; F(\mathbf{w}^*) \;\leq\; \left(1 - \frac{2m\epsilon_0 T_0}{(T_0+t)C}\right)\Big(\mathbb{E}\left[F(\mathbf{w}_t)\right] - F(\mathbf{w}^*)\Big) + \left(\frac{\epsilon_0 T_0}{T_0 + t}\right)^2 \frac{M S^2}{2c^2}. \tag{118}$$

Given that the product $2m\epsilon_0 T_0/C > 1$ as per the hypothesis, the sequence $\mathbb{E}\left[F(\mathbf{w}_{t+1})\right] - \; F(\mathbf{w}^*)$ satisfies the hypotheses of Lemma 8 with $a = 2m\epsilon_0 T_0/C$, $b = \epsilon_0^2 T_0^2 M S^2/2c^2$. It then follows from (103) and (104) that (35) is true for the C_0 constant defined in (36) upon identifying u_t with $\mathbb{E}\left[F(\mathbf{x}_{t+1})\right] - \; F(\mathbf{x}^*)$, C_0 with Q, and substituting $c = 2m\epsilon_0 T_0/C$, $b = \epsilon_0^2 T_0^2 M S^2/2c^2$ and $t_0 = T_0$ for their explicit values. ■

References

John R. Birge, Liqun Qi, and Xiaochun Chen. *A Stochastic Newton Method for Stochastic Quadratic Programs with Recourse*. School of Mathematics, University of New South Wales Sydney, Australia, 1994.

Antoine Bordes, Léon Bottou, and Patrick Gallinari. Sgd-qn: Careful quasi-newton stochastic gradient descent. *The Journal of Machine Learning Research*, 10:1737–1754, 2009.

Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pages 144–152. ACM, 1992.

Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010.

Léon Bottou and Yann Le Cun. On-line learning for very large data sets. *Applied Stochastic Models in Business and Industry*, 21(2):137–151, 2005.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge university press, 2004.

Charles G. Broyden, John E. Dennis, and Jorge J. Moré. On the local and superlinear convergence of quasi-newton methods. *IMA Journal of Applied Mathematics*, 12(3):223–245, 1973.

Richard H. Byrd, Jorge Nocedal, and Ya-Xiang Yuan. Global convergence of a class of quasi-newton methods on convex problems. *SIAM Journal on Numerical Analysis*, 24(5):1171–1190, 1987.

John E. Dennis and Jorge J. Moré. A characterization of superlinear convergence and its application to quasi-newton methods. *Mathematics of Computation*, 28(126):549–560, 1974.

Roger Fletcher. *Practical Methods of Optimization*. John Wiley & Sons, 2013.

Dong-Hui Li and Masao Fukushima. A modified bfgs method and its global convergence in nonconvex minimization. *Journal of Computational and Applied Mathematics*, 129(1):15–35, 2001.

Dong C. Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical Programming*, 45(1-3):503–528, 1989.

Aryan Mokhtari and Alejandro Ribeiro. Res: Regularized stochastic bfgs algorithm. *IEEE Transactions on Signal Processing*, 62:6089–6104, 2014a.

Aryan Mokhtari and Alejandro Ribeiro. A quasi-newton method for large scale support vector machines. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 8302–8306. IEEE, 2014b.

Andrew Y. Ng. Feature selection, l 1 vs. l 2 regularization, and rotational invariance. In *Proceedings of the Twenty-First International Conference on Machine Learning*, page 78. ACM, 2004.

Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer-Verlag, New York, NY, 2 edition, 1999.

Michael J. D. Powell. Some global convergence properties of a variable metric algorithm for minimization without exact line searches. *Nonlinear Programming*, 9:53–72, 1976.

Nicol N. Schraudolph, Jin Yu, and Simon Günter. A stochastic quasi-newton method for online convex optimization. In *International Conference on Artificial Intelligence and Statistics*, pages 436–443, 2007.

Shai Shalev-Shwartz and Nathan Srebro. Svm optimization: inverse dependence on training set size. In *Proceedings of the 25th International Conference on Machine Learning*, pages 928–935. ACM, 2008.

Shai Shalev-Shwartz, Yoram Singer, Nathan Srebro, and Andrew Cotter. Pegasos: Primal estimated sub-gradient solver for svm. *Mathematical programming*, 127(1):3–30, 2011.

Victor Solo and Xuan Kong. *Adaptive signal processing algorithms: stability and performance*. Prentice Hall, New Jersey, 1995.

G. Sun. Kdd cup track 2 soso. com ads prediction challenge, 2012. *Accessed August*, 1, 2012.

Peter Sunehag, Jochen Trumpf, Nicol N. Schraudolph, and Swaminathan V. N. Vishwanathan. Variable metric stochastic approximation theory. In *International Conference on Artificial Intelligence and Statistics*, pages 560–566, 2009.

Vladimir Vapnik. *The Nature of Statistical Learning Theory*. Springer Science & Business Media, 2000.

Michael Zargham, Alejandro Ribeiro, and Ali Jadbabaie. Accelerated backpressure algorithm. In *Global Communications Conference (GLOBECOM), 2013 IEEE*, pages 2269–2275. IEEE, 2013.

Tong Zhang. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *Proceedings of the Twenty-First International Conference on Machine Learning*, page 116. ACM, 2004.

Journal of Machine Learning Research 16 (2015) 3183-3217 Submitted 10/14; Revised 5/15; Published 12/15

On Semi-Supervised Linear Regression in Covariate Shift Problems

Kenneth Joseph Ryan KJRYAN@MAIL.WVU.EDU
Mark Vere Culp MVCULP@MAIL.WVU.EDU
Department of Statistics
West Virginia University
Morgantown, WV 26506, USA

Editor: Xiaotong Shen

Abstract

Semi-supervised learning approaches are trained using the full training (labeled) data and available testing (unlabeled) data. Demonstrations of the value of training with unlabeled data typically depend on a smoothness assumption relating the conditional expectation to high density regions of the marginal distribution and an inherent missing completely at random assumption for the labeling. So-called covariate shift poses a challenge for many existing semi-supervised or supervised learning techniques. Covariate shift models allow the marginal distributions of the labeled and unlabeled feature data to differ, but the conditional distribution of the response given the feature data is the same. An example of this occurs when a complete labeled data sample and then an unlabeled sample are obtained sequentially, as it would likely follow that the distributions of the feature data are quite different between samples. The value of using unlabeled data during training for the elastic net is justified geometrically in such practical covariate shift problems. The approach works by obtaining adjusted coefficients for unlabeled prediction which recalibrate the supervised elastic net to compromise: (i) maintaining elastic net predictions on the labeled data with (ii) shrinking unlabeled predictions to zero. Our approach is shown to dominate linear supervised alternatives on unlabeled response predictions when the unlabeled feature data are concentrated on a low dimensional manifold away from the labeled data and the true coefficient vector emphasizes directions away from this manifold. Large variance of the supervised predictions on the unlabeled set is reduced more than the increase in squared bias when the unlabeled responses are expected to be small, so an improved compromise within the bias-variance tradeoff is the rationale for this performance improvement. Performance is validated on simulated and real data.

Keywords: joint optimization, semi-supervised regression, usefulness of unlabeled data

1. Introduction

Semi-supervised learning is an active research area (Chapelle et al., 2006b; Zhu and Goldberg, 2009). Existing theoretical and empirical work typically invokes the missing completely at random (MCAR) assumption where the inclusion of a label is independent of the feature data and label. Under MCAR, there is theoretical work, mostly in classification, on finding borders that pass between dense regions of the data with particular emphasis on the cluster assumption (Chapelle et al., 2006b), semi-supervised smoothness assumptions (Lafferty and Wasserman, 2007; Azizyan et al., 2013), and manifold assumptions (Hein et al.,

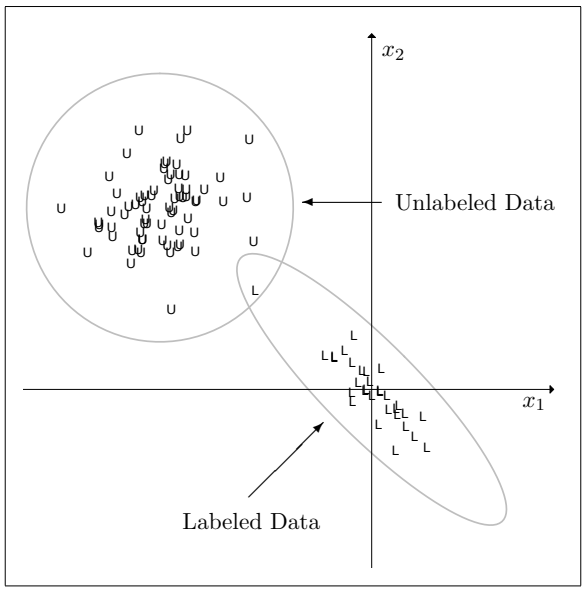


Figure 1: These feature data with $p = 2$ are referred to as the "block extrapolation" example because the unlabeled data "block" the 1st principal component of the labeled data. It is informative to think about how ridge regression would predict the unlabeled cases in this example. Favoring shrinking along the 2nd component will lead to high prediction variability. These block data are the primary working example throughout Sections 2-5, and it will be demonstrated that our semi-supervised approach has a clear advantage.

2005; Aswani et al., 2010). Many techniques including manifold regularization (Belkin et al., 2006) and graph cutting approaches (Wang et al., 2013) were developed to capitalize on unlabeled information during training, but beneath the surface of nearly all this work is the implicit or explicit use of MCAR (Lafferty and Wasserman, 2007).

Covariate shift is a different paradigm for semi-supervised learning (Moreno-Torres et al., 2008). It stipulates that the conditional distribution of the label given the feature data does not depend on the missingness of a label, but that the feature data distribution may depend on the missingness of a label. As a consequence, feature distributions can differ between labeled and unlabeled sets. Attempting to characterize smoothness assumptions between the regression function and the marginal of $\boldsymbol{X}$ (Azizyan et al., 2013) may not realize the value of unlabeled data if an implicit MCAR assumption breaks down. Instead, its value is in shrinking regression coefficients in an ideal direction to optimize the bias-variance tradeoff on unlabeled predictions. This is a novelty of our research direction.

The proposed approach is ideally suited for applications where the sequential generation of the labeled and unlabeled data causes covariate shift. Due to either matters of practicality or convenience the marginal distribution of the labeled feature data is likely to be profoundly different than that of the unlabeled feature data. Consider applications in drug discovery where the feature information consists of measurements on compounds and the responses are compound attributes, e.g., side effects of the drug, overall effect of the drug, or ability

to permeate the drug (Mente and Lombardo, 2005). Attributes can take years to obtain, while the feature information can be obtained much faster. As a result, the labeled data are often measurements on drugs with known attributes while the unlabeled data are usually compounds with unknown attributes that may potentially become new drugs (marketed to the public). Other applications mostly in classification include covariate shift problems (Yamazaki et al., 2007), reject inference problems from credit scoring (Moreno-Torres et al., 2008), spam filtering and brain computer interfacing (Sugiyama et al., 2007), and gene expression profiling of microarray data (Gretton et al., 2009). Gretton et al. (2009) further note that covariate shift occurs often in practice, but is under reported in the machine learning literature.

Many of the hypothetical examples to come do not conform to MCAR. The Figure 1 feature data are used to illustrate key concepts as they are developed in this work. Its labeled and unlabeled partitioning is unlikely if responses are MCAR. The vector of supervised ridge regression coefficients is proportionally shrunk more along the lower order principal component directions (Hastie et al., 2009). Such shrinking is toward a multiple of the unlabeled data centroid in the hypothetical Figure 1 scenario, so ridge regression may not deflate the variance of the unlabeled predictions enough. Standard methods for tuning parameter estimation via cross-validation do not account for the distribution of the unlabeled data either. Thus, supervised ridge regression is at a distinct disadvantage by not accounting for the unlabeled data during optimization. In general, the practical shortcoming of supervised regression (e.g., ridge, lasso, or elastic net) is to define regression coefficients that predict well for any unlabeled configuration. Our main contribution to come is a mathematical framework for adapting a supervised estimate to the unlabeled data configuration at hand for improved performance. It also provides interpretable "extrapolation" adjustments to the directions of shrinking as a byproduct.

Culp (2013) proposed a joint trained elastic net for semi-supervised regression under MCAR. The main idea was to use the joint training problem that encompasses the S^3VM (Chapelle et al., 2006a) and ψ-learning (Wang et al., 2009) to perform semi-supervised elastic net regression. The concept was that the unlabeled data should help with decorrelation and variable selection, two known hallmarks of the supervised elastic net extended to semi-supervised learning (Zou and Hastie, 2005). Culp (2013), however, did not contain a complete explanation of how exactly the approach used unlabeled data and under what set of mathematical assumptions it is expected to be useful.

The joint trained elastic net framework is strengthened in this paper to handle covariate shift. Rigorous geometrical and theoretical arguments are given for when it is expected to work. Circumstances where the feature data distribution changes by label status is the primary setting. One could view the unlabeled data as providing a group of extrapolations (or a separate manifold) from the labeled data. Even if responses are MCAR, the curse of dimensionality stipulates that nearly all predictions from a supervised learner are extrapolations in higher dimensions (Hastie et al., 2009), so the utility of the proposed semi-supervised approach is likely to increase with p.

Presentation of major concepts often begins with hypothetical, graphical examples in $p = 2$, but is followed by general mathematical treatments of $p \geq 2$. The work is written carefully so that themes extracted from $p = 2$ generalize. Section 2 provides a conceptual overview of the general approach with emphasis on the value of unlabeled data in covariate

shift before diving into the more rigorous mathematics in later sections. The problem is set-up formally in Section 3. The nature of regularization approaches (e.g., ridge, lasso, and elastic net) is studied with emphasis on a geometric perspective in Section 4. The geometry helps articulate realistic assumptions for the theoretical risk results in Section 5, and the theoretical risk results help define informative simulations and real data tests in Section 6. In addition, the simulations and real data applications validate the theoretical risk results. The combined effect is a characterization of when the approach is expected to outperform supervised alternatives in prediction. Follow-up discussion is in Section 7, and a proof for each proposition and theorem is in Appendix A.

2. The Value of Unlabeled Data due to Covariate Shift

The purpose of this section is to motivate the proposed approach for covariate shift data problems. The data are partitioned into the set of the labeled L and unlabeled U observations with $n = |L| + |U|$, and a response variable is recorded only for labeled observations. Let $\boldsymbol{Y}_L$ denote the observed $|L| \times 1$ vector of mean centered, labeled responses and $\boldsymbol{Y}_U$ the $|U| \times 1$ missing, unlabeled responses. If data are sorted by label status, the complete response vector and $n \times p$ model matrix partition to

$$\boldsymbol{Y} = \begin{pmatrix} \boldsymbol{Y}_L \\ \boldsymbol{Y}_U \end{pmatrix} \qquad \boldsymbol{X} = \begin{pmatrix} \boldsymbol{X}_L \\ \boldsymbol{X}_U \end{pmatrix}.$$

The $\boldsymbol{X}_L$ data are mean centered and standardized so that $\boldsymbol{X}_L^T \boldsymbol{X}_L$ is a correlation matrix, and $\boldsymbol{X}_U$ is also scaled using the means and variances of the labeled data. A supervised linear regression coefficient vector $\widehat{\boldsymbol{\beta}}^{(\text{SUP})}$ is trained using only the labeled data: $\boldsymbol{X}_L$ and $\boldsymbol{Y}_L$. Our semi-supervised $\widehat{\boldsymbol{\beta}}$ is trained with data $\boldsymbol{X}$ and $\boldsymbol{Y}_L$ by trading off: (i) supervised predictions $\boldsymbol{X}_L\widehat{\boldsymbol{\beta}} = \boldsymbol{X}_L\widehat{\boldsymbol{\beta}}^{(\text{SUP})}$ on L with (ii) shrinking $\boldsymbol{X}_U\widehat{\boldsymbol{\beta}}$ towards $\vec{0}$ on U, and the geometric value of this type of usage of the unlabeled data is presented in Section 2.1. A deeper presentation of this Section 2.1 concept is given by Sections 3 and 4. This work also demonstrates its theoretical performance under the standard linear model. In particular, the true coefficient vector must encourage shrinking as a good strategy in order for the unlabeled data to be useful in the proposed fashion. The introduction of this concept here in Section 2.2 precedes the corresponding mathematical presentation of performance bounds in Section 5.

2.1 Geometric Contribution of Unlabeled Data

The main strategy is to find a linear compromise between: (i) fully supervised prediction on the labeled data and (ii) predicting close to zero on the unlabeled data. Two examples of this are given below. In the "collinearity" example, it is possible to achieve both (i) and (ii). Thus, there is no need for a compromise. In the block extrapolation example, (i) and (ii) cannot be achieved simultaneously. The compromise is obtained by organizing the coefficient vector in terms of directions *orthogonal* to feature data extrapolation directions, so the predictions corresponding to more extreme unlabeled extrapolations are shrunk more.

Collinearity Example: Suppose $p = 2$, the two columns of labeled feature data are collinear with $\boldsymbol{X}_{L1} = \boldsymbol{X}_{L2}$, and the unlabeled data are also collinear and orthogonal to the labeled data with $\boldsymbol{X}_{U1} = -\boldsymbol{X}_{U2}$. The ordinary least squares estimator $\widehat{\boldsymbol{\beta}}^{(\text{OLS})}$ (i.e., a

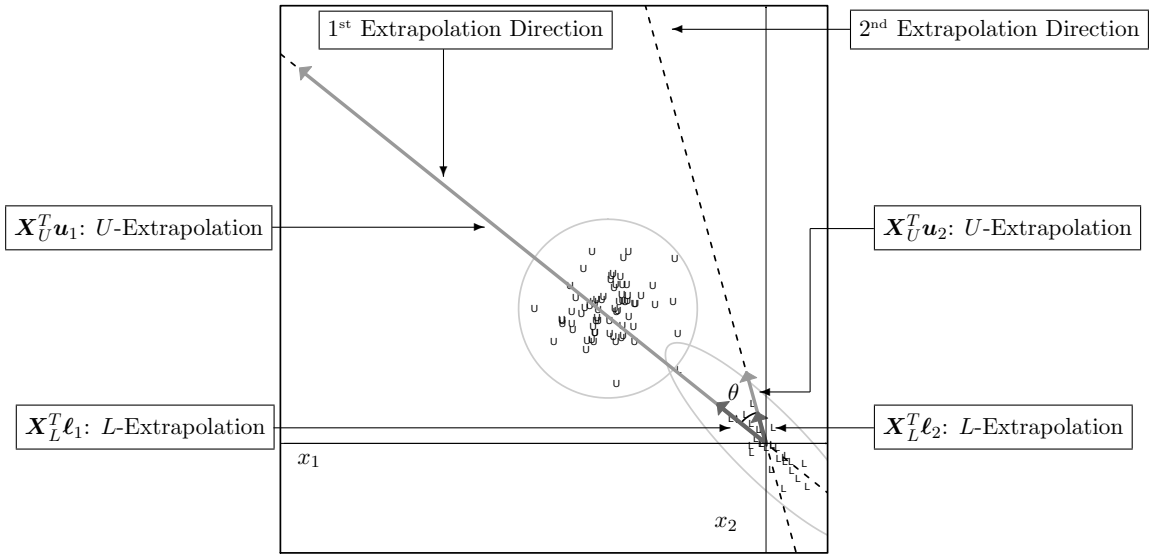


Figure 2: The dashed lines are the 1st and 2nd extrapolation directions for the block extrapolation example from Figure 1. The extent of U-extrapolation vector is a larger multiple of the extent of L-extrapolation vector in the 1st versus the 2nd extrapolation direction, so predictions corresponding to feature vectors on the 1st extrapolation direction are shrunk more than those on the 2nd extrapolation direction under the proposed method.

supervised linear regression estimator) is not unique since $\text{rank}(\boldsymbol{X}_L) = 1$, but the semi-supervised estimator $\widehat{\boldsymbol{\beta}} = \left(\boldsymbol{X}_{L1}^T\boldsymbol{Y}_L/2\right)\vec{1}$ is the unique solution to

$$\min_{\boldsymbol{\beta}} \|\boldsymbol{Y}_L - \boldsymbol{X}_L\boldsymbol{\beta}\|_2^2 + \|\boldsymbol{X}_U\boldsymbol{\beta}\|_2^2 . \tag{1}$$

This $\widehat{\boldsymbol{\beta}}$ is the ordinary least squares estimator with equal components, so it achieves objectives (i) $\boldsymbol{X}_L\widehat{\boldsymbol{\beta}} = \boldsymbol{X}_L\widehat{\boldsymbol{\beta}}^{(\text{OLS})}$ and (ii) $\boldsymbol{X}_U\widehat{\boldsymbol{\beta}} = \left(\boldsymbol{X}_{L1}^T\boldsymbol{Y}_L/2\right)\boldsymbol{X}_U\vec{1} = \vec{0}$. Optimization Problem (1) is a special case of the joint training framework to come in Section 3, and our general semi-supervised approach is based on this type of estimator.

Block Extrapolation Example: These data in Figure 2 include two lines marked as 1st and 2nd extrapolation directions, and each direction has extent vectors of largest U- and L-extrapolations ($\boldsymbol{X}_L^T\boldsymbol{\ell}_1$, $\boldsymbol{X}_U^T\boldsymbol{u}_1$ and $\boldsymbol{X}_L^T\boldsymbol{\ell}_2$, $\boldsymbol{X}_U^T\boldsymbol{u}_2$). Each L-based extent vector in Figure 2 is the longest possible of the form $\boldsymbol{X}_L^T\boldsymbol{\ell}$ in a given direction for $\boldsymbol{\ell} \in \mathbb{R}^{|L|}$ such that $\|\boldsymbol{\ell}\|_2^2 = 1$. Similarly, the U-based extent vectors are the longest possible in a given direction based on a unit length linear combination of the rows of $\boldsymbol{X}_U$. While precise mathematics on determining the two extrapolation directions is deferred until Section 4, it also turns out that the ratio of U- to L-extent vector lengths in the 2nd direction is never bigger than that in the 1st direction, i.e.,

$$\frac{\left\|\boldsymbol{X}_U^T\boldsymbol{u}_2\right\|_2}{\left\|\boldsymbol{X}_L^T\boldsymbol{\ell}_2\right\|_2} \leq \frac{\left\|\boldsymbol{X}_U^T\boldsymbol{u}_1\right\|_2}{\left\|\boldsymbol{X}_L^T\boldsymbol{\ell}_1\right\|_2}. \tag{2}$$

The sought after compromise is struck with semi-supervised estimator $\widehat{\boldsymbol{\beta}}$ by shrinking a supervised estimator $\widehat{\boldsymbol{\beta}}^{(\text{SUP})}$ with respect to a basis of directions orthogonal to the extrapolation directions. With this in mind, define the decomposition of a supervised estimate

$$\begin{aligned} &\widehat{\boldsymbol{\beta}}^{(\text{SUP})} = \tilde{\boldsymbol{\nu}}_1 + \tilde{\boldsymbol{\nu}}_2, \text{ where} \\ &\tilde{\boldsymbol{\nu}}_1 \text{ is orthogonal to the } 1^{\text{st}} \text{ extrapolation direction} \\ &\tilde{\boldsymbol{\nu}}_2 \text{ is orthogonal to the } 2^{\text{nd}} \text{ extrapolation direction,} \end{aligned} \tag{3}$$

and consider a semi-supervised estimate of the form

$$\widehat{\boldsymbol{\beta}} = p_1\tilde{\boldsymbol{\nu}}_1 + p_2\tilde{\boldsymbol{\nu}}_2, \text{ where } p_1 = \frac{\left\|\boldsymbol{X}_L^T\boldsymbol{\ell}_1\right\|_2}{\left\|\boldsymbol{X}_L^T\boldsymbol{\ell}_1\right\|_2 + \left\|\boldsymbol{X}_U^T\boldsymbol{u}_1\right\|_2} \text{ and } p_2 = \frac{\left\|\boldsymbol{X}_L^T\boldsymbol{\ell}_2\right\|_2}{\left\|\boldsymbol{X}_L^T\boldsymbol{\ell}_2\right\|_2 + \left\|\boldsymbol{X}_U^T\boldsymbol{u}_2\right\|_2}. \tag{4}$$

Coefficient shrinking is more focused on the vector orthogonal to the 1^{st} extrapolation direction because $0 \leq p_1 \leq p_2 \leq 1$ by Inequality (2).

A semi-supervised $\widehat{\boldsymbol{\beta}}$ from Display (4) was decomposed with regard to a basis orthogonal to directions of extrapolations from Display (3) so that linear predictions $\boldsymbol{x}_0^T\widehat{\boldsymbol{\beta}}$ at an arbitrary feature vector $\boldsymbol{x}_0 \in I\!R^2$ are shrunk more heavily when $\boldsymbol{x}_0$ is in directions with larger extrapolations. To demonstrate this, define a closely related decomposition of a feature vector

$$\begin{aligned} &\boldsymbol{x}_0 = \boldsymbol{\nu}_1 + \boldsymbol{\nu}_2, \text{ where} \\ &\boldsymbol{\nu}_1 \text{ is on the } 1^{\text{st}} \text{ extrapolation direction} \\ &\boldsymbol{\nu}_2 \text{ is on the } 2^{\text{nd}} \text{ extrapolation direction.} \end{aligned} \tag{5}$$

Together, Decompositions (4) and (5) result in the semi-supervised prediction

$$\boldsymbol{x}_0^T\widehat{\boldsymbol{\beta}} = p_1\boldsymbol{\nu}_1^T\tilde{\boldsymbol{\nu}}_2 + p_2\boldsymbol{\nu}_2^T\tilde{\boldsymbol{\nu}}_1$$

because $\boldsymbol{\nu}_1^T\tilde{\boldsymbol{\nu}}_1 = \boldsymbol{\nu}_2^T\tilde{\boldsymbol{\nu}}_2 = 0$ by construction. Thus, with fixed length feature vectors $\boldsymbol{x}_0 = \boldsymbol{\nu}_i$ on the 1^{st} and 2^{nd} extrapolation directions, the 1^{st} direction corresponds to a semi-supervised prediction $\boldsymbol{x}_0^T\widehat{\boldsymbol{\beta}}$ that is a more heavily shrunken version of its supervised prediction $\boldsymbol{x}_0^T\widehat{\boldsymbol{\beta}}^{(\text{SUP})}$ whenever $p_1 < p_2$.

The supervised estimate $\widehat{\boldsymbol{\beta}} = \widehat{\boldsymbol{\beta}}^{(\text{SUP})}$ results whenever $p_1 = p_2 = 1$, by Displays (3) and (4). Thus, supervised predictions are favored when L-based extrapolations $\left\|\boldsymbol{X}_L^T\boldsymbol{\ell}_i\right\|_2$ dominate U-based extrapolations $\left\|\boldsymbol{X}_U^T\boldsymbol{u}_i\right\|_2$ because $p_i \approx 1$ follows from Display (4). On the other hand, predictions near zero are favored when U-based extrapolations dominate L-based extrapolations ($p_i \approx 0$). In both cases, the p_i regulate the compromise (i) with (ii) for $\widehat{\boldsymbol{\beta}}$ term-by-term in each extrapolation direction. A significant contribution of this work is to provide a rigorous mathematical framework to study semi-supervised linear predictions for unlabeled extrapolations. In Section 4, directions of extrapolation and relative degrees of shrinking p_i are shown to follow from the joint trained optimization framework.

2.2 Model-based Contributions of Unlabeled Data

Under the linear model ($\mathbb{E}[\boldsymbol{Y}] = \boldsymbol{X}\boldsymbol{\beta}$ and $\mathbb{V}\text{ar}(\boldsymbol{Y}) = \sigma^2\boldsymbol{I}$), the coefficient parameter space partitions into lucky $(\boldsymbol{\beta}, \sigma^2)$ and unlucky $(\boldsymbol{\beta}, \sigma^2)$ subsets. Lucky versus unlucky $\boldsymbol{\beta}$ directions are not equally likely but depend greatly on the range and shape of the unlabeled

data manifold and on the model parameter σ^2. The general theme is that lucky (unlucky) $\boldsymbol{\beta}$'s are in directions orthogonal (parallel) to the unlabeled feature data manifold, so lower variability within this manifold implies more lucky $\boldsymbol{\beta}$ directions where our approach improves performance. A general bound is presented in Section 5 to help understand when our semi-supervised linear adjustment is guaranteed to outperform its supervised baseline on unlabeled predictions. Next, the collinearity and block extrapolation examples from Section 2.1 are revisited to illustrate lucky versus unlucky (or favorable versus unfavorable) prediction scenarios.

Collinearity Example: This example had $p = 2$, $\boldsymbol{X}_{L1} = \boldsymbol{X}_{L2}$, and $\boldsymbol{X}_{U1} = -\boldsymbol{X}_{U2}$. A lucky $\boldsymbol{\beta}$ follows with $\boldsymbol{\beta} = (b, b)^T$ for some arbitrary $b \in I\!R$, since $\boldsymbol{X}_U\boldsymbol{\beta} = \vec{0}$ is clearly ideal for the semi-supervised approach. On the other hand, suppose the true $\boldsymbol{\beta} = (b, -b)^T$ for some scalar b of large magnitude, and the components of $\boldsymbol{X}_{U1}$ are all of large magnitude with the same sign. This is an example of an unlucky $\boldsymbol{\beta}$ since the truth $\boldsymbol{X}_U\boldsymbol{\beta} = 2b\boldsymbol{X}_{U1}$ is far from the origin $\vec{0}$ with components of the same sign, so setting $\boldsymbol{X}_U\widehat{\boldsymbol{\beta}} = \vec{0}$ is less than ideal. Since $\boldsymbol{X}_L\boldsymbol{\beta} = \vec{0}$, the typical supervised linear regression estimators (e.g., ridge, lasso, and ENET) would predict the $\boldsymbol{X}_U$ cases close to $\vec{0}$ not $2b\boldsymbol{X}_{U1}$ and does not fair much better as a result. The bottom-line is that this unlucky $\boldsymbol{\beta}$ situation is not handled well by the conventional wisdom in machine learning of shrinking to optimize the bias-variance tradeoff (Hastie et al., 2009).

Block Extrapolation Example: This example was the block extrapolation from Figures 1 and 2. As it turns out, the ridge regression version of the Section 5 bound simplifies to a function of just $\boldsymbol{\beta}$ (call it $\sigma^2_{\text{LB}}(\boldsymbol{\beta})$) such that the semi-supervised approach is guaranteed to outperform the supervised approach whenever $\sigma^2 - \sigma^2_{\text{LB}}(\boldsymbol{\beta}) > 0$ at a given σ^2. Next, this bound is used to give a snapshot of parameter space $(\boldsymbol{\beta}, \sigma^2)$ in the context of the block extrapolation example, where lucky $\boldsymbol{\beta}$ correspond to $\sigma^2 - \sigma^2_{\text{LB}}(\boldsymbol{\beta}) > 0$ while unlucky $\boldsymbol{\beta}$ correspond to $\sigma^2 - \sigma^2_{\text{LB}}(\boldsymbol{\beta}) \leq 0$.

In order to investigate this, take all $\sigma^2 \in [0, 1]$ with all possible coefficient vectors

$$\boldsymbol{\beta}(\vartheta) = \begin{pmatrix} \sin(\vartheta) \\ \cos(\vartheta) \end{pmatrix} \text{ for } \vartheta \in [0, \pi]$$

on the right half of the unit circle. These parameters capture performance trends of an arbitrary fixed length $\boldsymbol{\beta}$ in all possible directions by the technical details in Section 5. Curves in Figure 3(a) are the bound $\sigma^2 - \sigma^2_{\text{LB}}(\boldsymbol{\beta}(\vartheta))$ as a function of ϑ at a given σ^2. Lighter (darker) curves correspond to smaller (larger) values σ^2 over an equally spaced grid on the interval $[0, 1]$, and the corresponding differences between unlabeled root mean-squared errors at the best supervised ($\text{RMSE}_U^{(\text{SUP})}$) and semi-supervised ($\text{RMSE}_U^{(\text{SEMI})}$) tuning parameter settings are provided in Figure 3(b). If ϑ is uniformly distributed on $[0, \pi]$, a lucky $\boldsymbol{\beta}$ is more likely than an unlucky $\boldsymbol{\beta}$, especially as σ^2 increases. The center for potentially large improvements in Figure 3(a) is roughly $\boldsymbol{\beta}(\pi/4) \approx (1, 1)^T/\sqrt{2}$. In addition, the unlabeled feature data centroid $\boldsymbol{X}_U^T\vec{1}/|U|$ in Figure 1 is roughly a multiple of $(-1, 1)^T$. Thus, $\vec{1}^T\boldsymbol{X}_U\boldsymbol{\beta}(\pi/4) \approx 0$. In other words, lucky $\boldsymbol{\beta}$ directions encourage shrinking predictions on U. On the other hand, unlucky $\boldsymbol{\beta}$ directions encourage large predictions. Take the center for little to no theoretically guaranteed improvement in Figure 3(a), i.e., $\boldsymbol{\beta}(3\pi/4) \approx (1, -1)^T/\sqrt{2}$. In this case, the true expected response at the unlabeled feature data centroid $\vec{1}^T\boldsymbol{X}_U\boldsymbol{\beta}(3\pi/4)/|U|$ is large because $\boldsymbol{\beta}(3\pi/4)$ is roughly a multiple of $\boldsymbol{X}_U^T\vec{1}/|U|$.

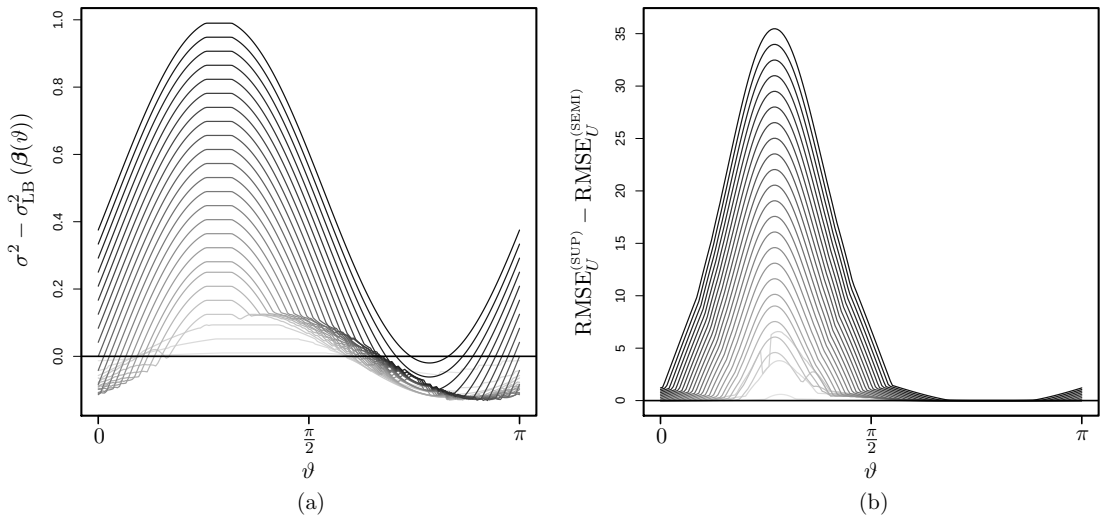


Figure 3: (a) The theoretical bound $\sigma^2 - \sigma^2_{\text{LB}}(\boldsymbol{\beta}(\vartheta))$ is plotted against ϑ for the block extrapolation example from Figures 1 and 2. Darker curves correspond to larger σ^2. Interest was in identifying ϑ such that $\sigma^2 - \sigma^2_{\text{LB}}(\boldsymbol{\beta}(\vartheta)) > 0$, since values greater than zero highlight the lucky unit length directions $\boldsymbol{\beta}(\vartheta)$ at a given σ^2 where our semi-supervised adjustment helps. (b) The corresponding differences between supervised and semi-supervised root mean squared errors (RMSEs) on the unlabeled set are displayed.

In general, the proposed approach is well suited for lucky $\boldsymbol{\beta}$ prediction problems, which include the following generalization of the Figure 1 block extrapolation example. The distance between feature data centroids (i.e., between the origin $\boldsymbol{X}_L^T\vec{1}/|L| = \vec{0}$ due to mean centering and $\boldsymbol{X}_U^T\vec{1}/|U|$) is increased relative to the variation about each centroid and the true coefficient vector $\boldsymbol{\beta}$ is not roughly a multiple of $\boldsymbol{X}_U^T\vec{1}/|U|$. One might conjecture lucky $\boldsymbol{\beta}$ to occur more often in practice during high-dimensional applications with large p by a sparsity of effects assumption (i.e., the true $\boldsymbol{\beta}$ has few non-zero components). For example, if the unlabeled feature data are concentrated on a low dimensional manifold away from the labeled data, there are more lucky directions for the true coefficient vector to emphasize directions away from the unlabeled feature data manifold. Also note that the supervised RMSEs are no better than semi-supervised in the block example, i.e., no negative differences in Figure 3(b). In theory, our technique handles unlucky $\boldsymbol{\beta}$ by defaulting to supervised predictions; see Remark 1 for how unlucky scenarios are handled empirically in practice.

Remark 1 *Nearly all supervised techniques would be challenged by an unlucky $\boldsymbol{\beta}$ direction since approaches typically improve predictive performance by shrinking (Hastie et al., 2009) and thus predicting large responses accurately on a covariate shifted data set is not what these techniques are designed to do. Supervised learning has a possible advantage over the proposed semi-supervised method in such situations by simply not shrinking extrapolation*

directions in the unlabeled data, but there is no guarantee here either (i.e., the supervised technique may still perform much worse). In this work, we do not assume that the response is generated under a lucky $\boldsymbol{\beta}$ linear model. Instead, a tuning parameter is used to move the semi-supervised estimator closer to supervised in such cases to mitigate the losses relative to supervised for an unlucky $\boldsymbol{\beta}$. Cross-validation is used to estimate this parameter in the results Section 6.

3. A Linear Joint Training Framework

The focus of this paper is the *joint trained elastic net*

$$\left(\widehat{\boldsymbol{\alpha}}_{\boldsymbol{\gamma},\boldsymbol{\lambda}}, \widehat{\boldsymbol{\beta}}_{\boldsymbol{\gamma},\boldsymbol{\lambda}}\right) = \underset{\boldsymbol{\alpha},\boldsymbol{\beta}}{\arg\min} \|\boldsymbol{Y}_L - \boldsymbol{X}_L\boldsymbol{\beta}\|_2^2 + \gamma_1\|\boldsymbol{X}_U(\boldsymbol{\alpha}-\boldsymbol{\beta})\|_2^2 + \gamma_1\gamma_2\|\boldsymbol{\alpha}\|_2^2 + \lambda_1\|\boldsymbol{\beta}\|_1^1 + \lambda_2\|\boldsymbol{\beta}\|_2^2, \tag{6}$$

where $\widehat{\boldsymbol{\beta}}_{\boldsymbol{\gamma},\boldsymbol{\lambda}}$ is appropriately scaled and $\boldsymbol{\lambda} = (\lambda_1, \lambda_2) \in [0,\infty]^2$ and $\boldsymbol{\gamma} = (\gamma_1, \gamma_2) \in [0,\infty]^2$ are tuning parameter vectors. The joint trained elastic net is an example of a joint training optimization framework used in semi-supervised learning (Chapelle et al., 2006b). Comparisons will be made to the *supervised optimization*

$$\widehat{\boldsymbol{\beta}}_{\boldsymbol{\lambda}}^{(\text{ENET})} = \underset{\boldsymbol{\beta}}{\arg\min} \|\boldsymbol{Y}_L - \boldsymbol{X}_L\boldsymbol{\beta}\|_2^2 + \lambda_1\|\boldsymbol{\beta}\|_1^1 + \lambda_2\|\boldsymbol{\beta}\|_2^2, \tag{7}$$

which is a partial solution to Joint Optimization (6) whenever $\gamma_1 = 0$ or $\gamma_2 = 0$.

Let $\boldsymbol{X}_U\boldsymbol{X}_U^T = \mathcal{O}_U\mathcal{D}_U\mathcal{O}_U^T$ be the eigendecomposition of this outer product and define

$$\boldsymbol{X}^{(\gamma_2)} \;=\; \begin{pmatrix} \boldsymbol{X}_L \\ \boldsymbol{X}_U^{(\gamma_2)} \end{pmatrix} = \begin{pmatrix} \boldsymbol{X}_L \\ \sqrt{\gamma_2}\left(\mathcal{D}_U + \gamma_2\boldsymbol{I}\right)^{-\frac{1}{2}}\mathcal{O}_U^T\boldsymbol{X}_U \end{pmatrix} \tag{8}$$

for $\gamma_2 > 0$. Proposition 2 establishes that the reduced problem

$$\widehat{\boldsymbol{\beta}}_{\boldsymbol{\gamma},\boldsymbol{\lambda}} = \underset{\boldsymbol{\beta}}{\arg\min} \|\boldsymbol{Y}_L - \boldsymbol{X}_L\boldsymbol{\beta}\|_2^2 + \gamma_1\left\|\boldsymbol{X}_U^{(\gamma_2)}\boldsymbol{\beta}\right\|_2^2 + \lambda_1\|\boldsymbol{\beta}\|_1^1 + \lambda_2\|\boldsymbol{\beta}\|_2^2 \tag{9}$$

is an alternative to Joint Optimization (6) over $(\boldsymbol{\alpha},\boldsymbol{\beta}) \in I\!R^p \times I\!R^p$.

Proposition 2 *If $\gamma_2 > 0$, then rank$(\boldsymbol{X}_U)$ = rank$\left(\boldsymbol{X}_U^{(\gamma_2)}\right)$ and a solution $\widehat{\boldsymbol{\beta}}_{\boldsymbol{\gamma},\boldsymbol{\lambda}}$ to Optimization Problem (9) is a partial solution to Optimization Problem (6).*

By Proposition 2, the semi-supervised estimate $\widehat{\boldsymbol{\beta}}_{\boldsymbol{\gamma},\boldsymbol{\lambda}}$ can be computed by an elastic net subroutine through data augmentation if the user simply inputs the supervised tuning parameters $\boldsymbol{\lambda}$ with model matrix $\left(\boldsymbol{X}_L^T,\ \sqrt{\gamma_1}\boldsymbol{X}_U^{(\gamma_2)^T}\right)^T$ and response vector $\left(\boldsymbol{Y}_L^T,\ \vec{0}^T\right)^T$ (i.e., impute $\boldsymbol{Y}_U = \vec{0}$). The Elastic Net Optimization Problem (7) is convex and can be solved quickly by the `glmnet` package in R (Friedman et al., 2010; R Core Team, 2015), so this helps make our semi-supervised adjustment computationally viable.

Matrix $\boldsymbol{X}_U^{(\gamma_2)^T}\boldsymbol{X}_U^{(\gamma_2)}$ from Optimization Problem (9) has the same eigenvectors as $\boldsymbol{X}_U^T\boldsymbol{X}_U$, but its eigenvalues homogenize to unity as $\gamma_2 \to 0$. As $\gamma_2 \to \infty$, $\boldsymbol{X}_U^{(\gamma_2)^T}\boldsymbol{X}_U^{(\gamma_2)} \to \boldsymbol{X}_U^T\boldsymbol{X}_U$, and Optimization Problem (9) goes to the *semi-supervised extreme*

$$\widehat{\boldsymbol{\beta}}_{(\gamma_1,\infty),\boldsymbol{\lambda}} = \underset{\boldsymbol{\beta}}{\arg\min} \|\boldsymbol{Y}_L - \boldsymbol{X}_L\boldsymbol{\beta}\|_2^2 + \gamma_1\|\boldsymbol{X}_U\boldsymbol{\beta}\|_2^2 + \lambda_1\|\boldsymbol{\beta}\|_1^1 + \lambda_2\|\boldsymbol{\beta}\|_2^2. \tag{10}$$

Semi-Supervised Extreme (10) with $\boldsymbol{\lambda} = \vec{0}$ and $\gamma_1 = 1$ was seen earlier in Problem (1) during the conceptual overview. Finite $\gamma_2 > 0$ will later be seen to produce intermediate compromises between Supervised (7) and Semi-Supervised Extreme (10).

4. Geometry of Semi-Supervised Linear Regression

A geometrical understanding of the Joint Trained Elastic Net (6) is developed through the following logical progression: Section 4.1 joint trained least squares $\boldsymbol{\lambda} = \vec{0}$, Section 4.2 joint trained ridge $\boldsymbol{\lambda} = (0, \lambda_2)$, Section 4.3 joint trained lasso $\boldsymbol{\lambda} = (\lambda_1, 0)$, and then Section 4.4 joint trained elastic net regression $\boldsymbol{\lambda}$. Last, Section 4.5 provides a gallery of geometrical examples. The conceptual overview from Section 2.1 lines-up closely with the mathematics of Section 4.1 and is back-referenced extensively to help the reader make connections. The ridge, lasso, and elastic net semi-supervised geometries do, to some degree, simply follow from their well-known supervised properties when combined with the geometrical properties of joint trained (semi-supervised) least squares. However, an important subtlety is worth mentioning. This geometry section, especially Sections 4.3 and 4.4, establishes properties of the Joint Trained Elastic Net (6), and these properties are stated as the assumptions of Section 5 in order to derive general performance bounds that necessarily apply to the joint trained elastic net.

4.1 Joint Trained Least Squares

Optimization Problem (9) with $\boldsymbol{\lambda} = \vec{0}$ reduces to *joint trained least squares*

$$\widehat{\boldsymbol{\beta}}_{\boldsymbol{\gamma}} = \arg\min_{\boldsymbol{\beta}} \|\boldsymbol{Y}_L - \boldsymbol{X}_L\boldsymbol{\beta}\|_2^2 + \gamma_1 \left\| \boldsymbol{X}_U^{(\gamma_2)}\boldsymbol{\beta} \right\|_2^2. \tag{11}$$

Briefly recall the collinearity example from Section 2.1, i.e., $p = 2$, $\boldsymbol{X}_{L1} = \boldsymbol{X}_{L2}$, $\boldsymbol{X}_{U1} = -\boldsymbol{X}_{U2}$, and $\boldsymbol{\gamma} = (1, \infty)$. A supervised $\widehat{\boldsymbol{\beta}}^{(\text{OLS})}$ was not unique, but the $\widehat{\boldsymbol{\beta}}^{(\text{OLS})}$ with equal components was the unique semi-supervised Estimator (11). In general, Estimator (11) is unique whenever $\boldsymbol{\gamma} > \vec{0}$ and $\text{rank}(\boldsymbol{X}) = p$. Henceforth, assume $\text{rank}(\boldsymbol{X}_L) = p$, so $\widehat{\boldsymbol{\beta}}^{(\text{OLS})} = \left(\boldsymbol{X}_L^T\boldsymbol{X}_L\right)^{-1}\boldsymbol{X}_L^T\boldsymbol{Y}_L$ is unique during this discussion of joint trained least squares. Section 4.2 on joint trained ridge regression is tailored for $\text{rank}(\boldsymbol{X}_L) < p$.

Figure 4(a) displays the semi-supervised extreme $\widehat{\boldsymbol{\beta}}_{\gamma_1,\infty}$ from the block extrapolation example for a particular $\gamma_1 > 0$ based on the calculus of Lagrangian multipliers. For general $p \geq 2$ with $\gamma_2 \geq 0$, there exists unique scalars $a_{\gamma_2}, b_{\gamma_2}$ such that the ellipsoids

$$\boldsymbol{\beta}^T {\boldsymbol{X}_U^{(\gamma_2)}}^T \boldsymbol{X}_U^{(\gamma_2)}\boldsymbol{\beta} \leq a_{\gamma_2} \tag{12}$$

$$\left(\boldsymbol{\beta} - \widehat{\boldsymbol{\beta}}^{(\text{OLS})}\right)^T \boldsymbol{X}_L^T\boldsymbol{X}_L \left(\boldsymbol{\beta} - \widehat{\boldsymbol{\beta}}^{(\text{OLS})}\right) \geq b_{\gamma_2} \tag{13}$$

have the same tangent slope at the point of intersection $\widehat{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}$. A novelty of the semi-supervised approach, that holds for general $p \geq 2$, is the use of origin-centered Ellipsoids (12) as opposed to the multidimensional spheres used in supervised ridge regression.

When $\gamma_2 \approx 0$, $\widehat{\boldsymbol{\beta}}_{\boldsymbol{\gamma}} \approx \widehat{\boldsymbol{\beta}}_{\gamma_1}^{(\text{RIDGE})} = \left(\boldsymbol{X}_L^T\boldsymbol{X}_L + \gamma_1\boldsymbol{I}\right)^{-1}\boldsymbol{X}_L^T\boldsymbol{Y}_L$ because Ellipsoids (12) are roughly spherical. When γ_2 is large, $\widehat{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}$ approximates a point on the semi-supervised

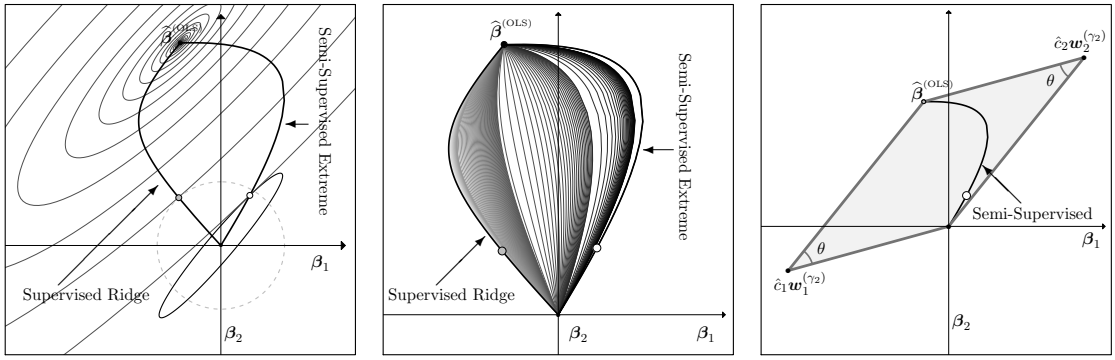


Figure 4: The Figure 1 block example is revisited. (a) A labeled response $\boldsymbol{Y}_L$ that resulted in the plotted estimate $\widehat{\boldsymbol{\beta}}^{(\mathrm{OLS})}$ is part of the assumed labeled data set. Each estimate on the semi-supervised extreme $\widehat{\boldsymbol{\beta}}_{\gamma_1,\infty}$, like the small white circle at $\gamma_1 = 0.18$, is the intersection of an origin-center Ellipse (12) and a $\widehat{\boldsymbol{\beta}}^{(\mathrm{OLS})}$-centered Ellipse (13) having the same tangent slope at this point of intersection. Similarly, each ridge estimate, like the small gray circle with $\lambda_2 = 5.9$, uses origin-centered, concentric circles instead of Ellipses (12). (b) Paths $\widehat{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}$ varying γ_1 with darker curves for larger γ_2 fill-in all possible compromises between supervised ridge and the semi-supervised extreme. (c) The semi-supervised extreme $\widehat{\boldsymbol{\beta}}_{\gamma_1,\infty}$ is shrunk within its bounding parallelogram from supervised $\widehat{\boldsymbol{\beta}}^{(\mathrm{OLS})}$ toward the origin as $\gamma_1 \to \infty$.

extreme. For example, take the point along the supervised ridge (semi-supervised extreme) path indicated by the small gray (white) circle in Figure 4. Paths $\widehat{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}$, like those in Figure 4(b), start at $\widehat{\boldsymbol{\beta}}^{(\mathrm{OLS})}$ and converge to a point in the null space of $\boldsymbol{X}_U$ as $\gamma_1 \to \infty$.

The semi-supervised estimator for any $\boldsymbol{\gamma}$ is

$$\begin{aligned} \widehat{\boldsymbol{\beta}}_{\boldsymbol{\gamma}} &= \left(\boldsymbol{X}_L^T\boldsymbol{X}_L + \gamma_1 \boldsymbol{X}_U^{(\gamma_2)^T}\boldsymbol{X}_U^{(\gamma_2)}\right)^{-1}\boldsymbol{X}_L^T\boldsymbol{X}_L\widehat{\boldsymbol{\beta}}^{(\mathrm{OLS})} \\ &= \left(\boldsymbol{I} + \gamma_1 \boldsymbol{M}^{(\gamma_2)}\right)^{-1}\widehat{\boldsymbol{\beta}}^{(\mathrm{OLS})}, \text{ where } \boldsymbol{M}^{(\gamma_2)} = \left(\boldsymbol{X}_L^T\boldsymbol{X}_L\right)^{-1}\boldsymbol{X}_U^{(\gamma_2)^T}\boldsymbol{X}_U^{(\gamma_2)}. \end{aligned} \tag{14}$$

An eigenbasis $\left\{\left(\boldsymbol{w}_i^{(\gamma_2)}, \tau_i^{(\gamma_2)}\right)\right\}_{i=1}^p$ of $\boldsymbol{M}^{(\gamma_2)}$ such that $\left\|\boldsymbol{X}_L\boldsymbol{w}_i^{(\gamma_2)}\right\|_2^2 = 1$ will be used to help understand how joint trained least squares regression coefficients are shrunk. Proposition 3 establishes that this important eigenbasis is real whether or not matrix $\boldsymbol{M}^{(\gamma_2)}$ is symmetric.

Proposition 3 *Any eigenbasis of the possibly non-symmetric matrix $\boldsymbol{M}^{(\gamma_2)}$ is real with eigenvalues $\tau_1^{(\gamma_2)} \geq \cdots \geq \tau_p^{(\gamma_2)} \geq 0$. Furthermore, $\tau_i^{(\gamma_2)} = 0$ iff $i > \mathrm{rank}(\boldsymbol{X}_U)$.*

While $\left\{\boldsymbol{w}_i^{(\gamma_2)}\right\}_{i=1}^p$ may be neither orthogonal nor unit length,

$$\widehat{\boldsymbol{\beta}}^{(\mathrm{OLS})} = \hat{c}_1^{(\gamma_2)}\boldsymbol{w}_1^{(\gamma_2)} + \cdots + \hat{c}_p^{(\gamma_2)}\boldsymbol{w}_p^{(\gamma_2)} \tag{15}$$

for some scalars $\hat{c}_i^{(\gamma_2)}$, and by Equations (14) and (15),

$$\widehat{\boldsymbol{\beta}}_{\boldsymbol{\gamma}} = \left(\frac{1}{1+\gamma_1\tau_1^{(\gamma_2)}}\right)\hat{c}_1^{(\gamma_2)}\boldsymbol{w}_1^{(\gamma_2)} + \cdots + \left(\frac{1}{1+\gamma_1\tau_p^{(\gamma_2)}}\right)\hat{c}_p^{(\gamma_2)}\boldsymbol{w}_p^{(\gamma_2)}. \tag{16}$$

Equations (15) and (16) generalize Estimator (4) from Section 2.1 to $p \geq 2$. The terms on the right of Equation (15) were previously denoted by the $\tilde{\boldsymbol{\nu}}_i$ from Display (3), and these terms are weighted by proportions on the right of Equation (16) that were previously denoted by the p_i from Display (4). Eigenvector $\hat{c}_1^{(\gamma_2)}\boldsymbol{w}_1^{(\gamma_2)}$ is proportionally shrunk the most at any fixed $\gamma_1 > 0$ because its proportion weight $1/\left(1+\gamma_1\tau_1^{(\gamma_2)}\right)$ is the smallest.

The bounding parallelogram in Figure 4(c) helps introduce another interpretation of Equation (16). This parallelogram has opposite corners at the origin and $\widehat{\boldsymbol{\beta}}^{(\text{OLS})}$ and sides parallel to the eigenvectors of $\boldsymbol{M}^{(\gamma_2)}$. The path $\widehat{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}$ shrinks from $\widehat{\boldsymbol{\beta}}^{(\text{OLS})}$ to the origin along the sides with corner $\hat{c}_2^{(\gamma_2)}\boldsymbol{w}_2^{(\gamma_2)}$ as $\gamma_1 \in [0,\infty]$ increases and does so more closely when $\tau_1^{(\gamma_2)}$ and $\tau_2^{(\gamma_2)}$ differ in magnitude. Proposition 4 generalizes this concept to arbitrary $\gamma_2 \geq 0$ and $p \geq 2$.

Proposition 4 *The path $\widehat{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}$ as a function of $\gamma_1 \geq 0$ is bounded within a p-dimensional parallelotope with corners at each binary linear combination of $\left\{\hat{c}_1^{(\gamma_2)}\boldsymbol{w}_1^{(\gamma_2)}, \ldots, \hat{c}_p^{(\gamma_2)}\boldsymbol{w}_p^{(\gamma_2)}\right\}$. Furthermore, the terminal point as $\gamma_1 \to \infty$ is the corner $\sum_{i=1}^{p}\mathcal{I}_{\{i>rank(\boldsymbol{X}_U)\}}\hat{c}_i^{(\gamma_2)}\boldsymbol{w}_i^{(\gamma_2)}$ with indicator $\mathcal{I}_{\{\cdot\}}$.*

The conceptual overview in Section 2.1 made a careful distinction between shrinking regression coefficients $\widehat{\boldsymbol{\beta}}$ versus shrinking linear predictions $\boldsymbol{x}_0^T\widehat{\boldsymbol{\beta}}$. Vectors $\tilde{\boldsymbol{\nu}}_i$ from Display (3) were related to coefficient shrinking, whereas $\boldsymbol{\nu}_i$ from Display (5) were the feature vectors $\boldsymbol{x}_0$ related to prediction shrinking. Mathematically, eigenvectors $\boldsymbol{w}_i^{(\gamma_2)}$ determine directions of coefficient shrinking. Since $p = 2$, the Section 2.1 discussion in-fact concentrated on all feature vectors $\boldsymbol{w}_1^{(\gamma_2)^\perp}$ and $\boldsymbol{w}_2^{(\gamma_2)^\perp}$, and an eigenvector direction of maximum (minimum) coefficient shrinking was orthogonal to feature vectors of maximum (minimum) prediction shrinking. Generalizing this story to $p > 2$ also results in p directions of coefficient shrinking and p feature vector directions of interpretable prediction shrinking, but the mathematics has the following subtlety. When $p > 2$, a direction of coefficient shrinking $\boldsymbol{w}_i^{(\gamma_2)}$ is orthogonal to a $p-1$ dimensional vector space $\boldsymbol{w}_i^{(\gamma_2)^\perp}$ of feature vectors, so if $p-1 \geq 2$, vector space $\boldsymbol{w}_1^{(\gamma_2)^\perp}$ consists of an infinite number of directions. Proposition 5 below provides a convenient form for the line in common to all $\boldsymbol{w}_j^{(\gamma_2)^\perp}$ with $j \neq i$ for each $i \in \{1,\ldots,p\}$ by establishing a relationship between $\boldsymbol{w}_i^{(\gamma_2)}$, $\boldsymbol{w}_i^{(\gamma_2)^\perp}$, and $\boldsymbol{X}^{(\gamma_2)}$ from Equation (8). These p lines of feature data vectors for arbitrary $p \geq 2$ will later be seen to have a clear interpretation when it comes to prediction shrinking, so we call them *extrapolation directions*.

Proposition 5 *The* $\text{span}\left(\boldsymbol{X}^{(\gamma_2)^T}\boldsymbol{X}^{(\gamma_2)}\boldsymbol{w}_i^{(\gamma_2)}\right) = \bigcap_{j\in\{1,\ldots,p\}-\{i\}}\boldsymbol{w}_j^{(\gamma_2)^\perp}$ $\forall i \in \{1,\ldots,p\}$. *Henceforth, the line* $\text{span}\left(\boldsymbol{X}^{(\gamma_2)^T}\boldsymbol{X}^{(\gamma_2)}\boldsymbol{w}_i^{(\gamma_2)}\right)$ *is called the i^{th} extrapolation direction $\forall i \in \{1,\ldots,p\}$.*

The i^{th} extrapolation direction necessarily traces out a line because it's all scalar multiples of the nonzero vector $\boldsymbol{X}^{(\gamma_2)^T}\boldsymbol{X}^{(\gamma_2)}\boldsymbol{w}_i^{(\gamma_2)}$. Any feature vector on the i^{th} *extrapolation direction*, i.e., $\boldsymbol{x}_0 \in \bigcap_{j\in\{1,\ldots,p\}-\{i\}} \boldsymbol{w}_j^{(\gamma_2)^\perp}$ from Proposition 5, is of special note. Their Equation (16) semi-supervised predictions simplify to $\boldsymbol{x}_0^T\widehat{\boldsymbol{\beta}}_{\boldsymbol{\gamma}} = \hat{c}_i^{(\gamma_2)}/\left(1+\gamma_1\tau_i^{(\gamma_2)}\right)\boldsymbol{x}_0^T\boldsymbol{w}_i^{(\gamma_2)}$ and are shrunk more (relative to the corresponding OLS supervised prediction $\boldsymbol{x}_0^T\widehat{\boldsymbol{\beta}}^{(\text{OLS})} = \hat{c}_i^{(\gamma_2)}\boldsymbol{x}_0^T\boldsymbol{w}_i^{(\gamma_2)}$) for smaller $i \in \{1,\ldots,p\}$ at any fixed $\gamma_1 > 0$ because $\tau_1^{(\gamma_2)} \geq \cdots \geq \tau_p^{(\gamma_2)}$.

Next, the i^{th} extrapolation direction is shown to be one of more (or less) extreme unlabeled extrapolations. With this in mind, use the indicator function $\mathcal{I}_{\{\cdot\}}$ to define the positive number $\kappa_i^{(\gamma_2)} = \tau_i^{(\gamma_2)} + \mathcal{I}_{\{i>\text{rank}(\boldsymbol{X}_U)\}}$ and define the vectors

$$\boldsymbol{\ell}_i^{(\gamma_2)} = \boldsymbol{X}_L\boldsymbol{w}_i^{(\gamma_2)} \text{ and } \boldsymbol{u}_i^{(\gamma_2)} = \frac{\boldsymbol{X}_U^{(\gamma_2)}\boldsymbol{w}_i^{(\gamma_2)}}{\sqrt{\kappa_i^{(\gamma_2)}}}. \tag{17}$$

Vectors (17) in the semi-supervised extreme of $\gamma_2 = \infty$ were temporarily denoted by $\boldsymbol{\ell}_i$ and $\boldsymbol{u}_i$ during their more conceptual introduction within Section 2.1 (e.g., Figure 2). It was also stated previously during this overview that $\boldsymbol{\ell}_i$ and $\boldsymbol{u}_i$ were unit length. Proposition 6 is a generalization.

Proposition 6 *If $\gamma_2 > 0$, vectors $\left\{\boldsymbol{\ell}_i^{(\gamma_2)}\right\}_{i=p}^{1}$ and $\left\{\boldsymbol{u}_i^{(\gamma_2)}\right\}_{i=1}^{rank(\boldsymbol{X}_U)}$ are orthonormal bases for the column spaces of $\boldsymbol{X}_L$ and $\boldsymbol{X}_U^{(\gamma_2)}$, and $\boldsymbol{u}_i^{(\gamma_2)} = \vec{0}$ if $i > rank(\boldsymbol{X}_U)$.*

Section 2.1 also introduced extents of L- and U-extrapolation. Vectors (17) are used to define these now for each $i \in \{1,\ldots,p\}$ as

$$\begin{aligned} \boldsymbol{X}_L^T\boldsymbol{\ell}_i^{(\gamma_2)} &\text{ Extent of } L\text{-Extrapolation (in the } i^{\text{th}} \text{ Direction)} \\ \boldsymbol{X}_U^{(\gamma_2)^T}\boldsymbol{u}_i^{(\gamma_2)} &\text{ Extent of } U\text{-Extrapolation (in the } i^{\text{th}} \text{ Direction), where} \\ \text{span}\left(\boldsymbol{X}^{(\gamma_2)^T}\boldsymbol{X}^{(\gamma_2)}\boldsymbol{w}_i^{(\gamma_2)}\right) &\text{ is the } i^{\text{th}} \text{ Direction of Extrapolation from Proposition 5.} \end{aligned} \tag{18}$$

Propositions 7 establishes that the i^{th} extent vectors are in-fact on the i^{th} extrapolation direction.

Proposition 7 *For each $i \in \{1,\ldots,p\}$,*

$$\begin{aligned} \boldsymbol{X}_L^T\boldsymbol{\ell}_i^{(\gamma_2)} &= \frac{1}{1+\tau_i^{(\gamma_2)}}\boldsymbol{X}^{(\gamma_2)^T}\boldsymbol{X}^{(\gamma_2)}\boldsymbol{w}_i^{(\gamma_2)} \\ \boldsymbol{X}_U^{(\gamma_2)^T}\boldsymbol{u}_i^{(\gamma_2)} &= \frac{\tau_i^{(\gamma_2)}}{(1+\tau_i^{(\gamma_2)})\sqrt{\kappa_i^{(\gamma_2)}}}\boldsymbol{X}^{(\gamma_2)^T}\boldsymbol{X}^{(\gamma_2)}\boldsymbol{w}_i^{(\gamma_2)}, \end{aligned}$$

so $\boldsymbol{X}^{(\gamma_2)^T}\boldsymbol{X}^{(\gamma_2)}\boldsymbol{w}_i^{(\gamma_2)}$, $\boldsymbol{X}_L^T\boldsymbol{\ell}_i^{(\gamma_2)}$, and $\boldsymbol{X}_U^{(\gamma_2)^T}\boldsymbol{u}_i^{(\gamma_2)}$ are parallel vectors in $\mathbb{R}^p$.

Previously defined vectors are now verified to possess fundamental interpretations: (i) Extent Vectors (18) do indeed measure "extrapolation extents" in a sensible manner, (ii) Vectors (17) determine shrinking directions for joint trained least squares fits $\boldsymbol{X}\widehat{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}$, and (iii) magnitudes of extent vectors regulate the shrinking of regression coefficients $\widehat{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}$. These three interpretations are gleaned by applying Propositions 6 and 7 in conjunction with well-known properties of orthogonal projection matrices and quadratic forms from linear algebra. The $n \times p$ matrix identity

$$\boldsymbol{X}^{(\gamma_2)} \begin{pmatrix} \boldsymbol{w}_1^{(\gamma_2)} & \cdots & \boldsymbol{w}_p^{(\gamma_2)} \end{pmatrix} = \left(\begin{pmatrix} \boldsymbol{\ell}_1^{(\gamma_2)} \\ \sqrt{\kappa_1^{(\gamma_2)}}\boldsymbol{u}_1^{(\gamma_2)} \end{pmatrix} \cdots \begin{pmatrix} \boldsymbol{\ell}_p^{(\gamma_2)} \\ \sqrt{\kappa_p^{(\gamma_2)}}\boldsymbol{u}_p^{(\gamma_2)} \end{pmatrix} \right) \tag{19}$$

follows from Definitions (17). The right of Equation (19) has orthogonal columns by Proposition 6, and the columns on the left of Equation (19) are eigenvectors with eigenvalue one of the orthogonal projection matrix $\boldsymbol{X}^{(\gamma_2)} \left(\boldsymbol{X}^{(\gamma_2)^T} \boldsymbol{X}^{(\gamma_2)} \right)^{-1} \boldsymbol{X}^{(\gamma_2)^T}$. Therefore, the columns of Matrix (19) are an orthogonal basis for the eigenspace of $\boldsymbol{X}^{(\gamma_2)} \left(\boldsymbol{X}^{(\gamma_2)^T} \boldsymbol{X}^{(\gamma_2)} \right)^{-1} \boldsymbol{X}^{(\gamma_2)^T}$ corresponding to eigenvalue one, because $\text{rank}\left(\boldsymbol{X}^{(\gamma_2)} \right) = p$ is a necessary condition for the joint trained least squares assumption that $\text{rank}(\boldsymbol{X}_L) = p$.

Projection matrix $\boldsymbol{X}^{(\gamma_2)} \left(\boldsymbol{X}^{(\gamma_2)^T} \boldsymbol{X}^{(\gamma_2)} \right)^{-1} \boldsymbol{X}^{(\gamma_2)^T}$ is nonnegative definite, so its main diagonal block sub matrices based on the L, U data partition are also nonnegative definite. The nonnegative definite, rank-p, sub matrix $\boldsymbol{X}_L \left(\boldsymbol{X}^{(\gamma_2)^T} \boldsymbol{X}^{(\gamma_2)} \right)^{-1} \boldsymbol{X}_L^T$ has orthonormal eigenvectors $\left\{ \boldsymbol{\ell}_i^{(\gamma_2)} \right\}_{i=p}^{1}$ corresponding to its nonzero eigenvalues $1/(1+\tau_i^{(\gamma_2)})$ by Propositions 6 and 7. Similarly, nonnegative definite sub matrix $\boldsymbol{X}_U^{(\gamma_2)} \left(\boldsymbol{X}^{(\gamma_2)^T} \boldsymbol{X}^{(\gamma_2)} \right)^{-1} \boldsymbol{X}_U^{(\gamma_2)^T}$ has orthonormal eigenvectors $\left\{ \boldsymbol{u}_i^{(\gamma_2)} \right\}_{i=1}^{\text{rank}(\boldsymbol{X}_U)}$ corresponding to its nonzero eigenvalues $\tau_i^{(\gamma_2)}/(1+\tau_i^{(\gamma_2)})$. Well-known eigenvector solutions to constrained optimizations of quadratic forms imply

$$\begin{aligned} \boldsymbol{\ell}_i^{(\gamma_2)} &= \underset{\boldsymbol{v} \in I\!R^{|L|}: \boldsymbol{v}^T\boldsymbol{v}=1, \boldsymbol{v}^T\boldsymbol{\ell}_j^{(\gamma_2)}=0 \ \forall j>i}{\arg\max} \boldsymbol{v}^T \boldsymbol{X}_L \left(\boldsymbol{X}^{(\gamma_2)^T} \boldsymbol{X}^{(\gamma_2)} \right)^{-1} \boldsymbol{X}_L^T \boldsymbol{v} \\ \boldsymbol{u}_i^{(\gamma_2)} &= \underset{\boldsymbol{v} \in I\!R^{|U|}: \boldsymbol{v}^T\boldsymbol{v}=1, \boldsymbol{v}^T\boldsymbol{u}_j^{(\gamma_2)}=0 \ \forall j<i}{\arg\max} \boldsymbol{v}^T \boldsymbol{X}_U^{(\gamma_2)} \left(\boldsymbol{X}^{(\gamma_2)^T} \boldsymbol{X}^{(\gamma_2)} \right)^{-1} \boldsymbol{X}_U^{(\gamma_2)^T} \boldsymbol{v}. \end{aligned}$$

In other words, the unit length weight vectors on the rows of $\boldsymbol{X}_L$ (of $\boldsymbol{X}_U^{(\gamma_2)}$) that maximize a Mahalanobis distance measuring extent of extrapolation subject to orthogonality constraints are the eigenvectors $\left\{ \boldsymbol{\ell}_i^{(\gamma_2)} \right\}_{i=p}^{1}$ (eigenvectors $\left\{ \boldsymbol{u}_i^{(\gamma_2)} \right\}_{i=1}^{\text{rank}(\boldsymbol{X}_U)}$) sorted by descending positive eigenvalues. Proposition 7 also establishes that each eigenvalue

$$\tau_i^{(\gamma_2)} = \frac{\left\| \boldsymbol{X}_U^{(\gamma_2)^T} \boldsymbol{u}_i^{(\gamma_2)} \right\|_2}{\left\| \boldsymbol{X}_L^T \boldsymbol{\ell}_i^{(\gamma_2)} \right\|_2} \tag{20}$$

of the shrinking matrix $\boldsymbol{M}^{(\gamma_2)}$ from Display (14) is a ratio of parallel extent eigenvector lengths, so the extent of U-extrapolation is larger (smaller) than the corresponding L-extent in the i^{th} direction of extrapolation if $\tau_i^{(\gamma_2)} > 1$ (if $\tau_i^{(\gamma_2)} < 1$).

The joint trained least squares fits vector for all n observations has the form

$$\boldsymbol{X}\widehat{\boldsymbol{\beta}}_{\boldsymbol{\gamma}} = \sum_{i=1}^{p} \hat{c}_i^{(\gamma_2)} \left(\frac{1}{1+\gamma_1\tau_i^{(\gamma_2)}} \right) \begin{pmatrix} \boldsymbol{\ell}_i^{(\gamma_2)} \\ \sqrt{\kappa_i^{(\gamma_2)}/\gamma_2}\mathcal{O}_U \left(\mathcal{D}_U + \gamma_2 \boldsymbol{I}\right)^{\frac{1}{2}} \boldsymbol{u}_i^{(\gamma_2)} \end{pmatrix}$$

by Equations (16) and (19) and the reverse of Transformation (8). Thus, eigenvectors $\boldsymbol{\ell}_i^{(\gamma_2)}$ and $\boldsymbol{u}_i^{(\gamma_2)}$ involved in constructing the i^{th} extrapolation direction with smaller $i \in \{1, \ldots, p\}$ are used to shrink fits more as γ_1 is increased. By Equation (16) and Ratios (20), coefficient vector

$$\widehat{\boldsymbol{\beta}}_{\boldsymbol{\gamma}} = \sum_{i=1}^{p} \left(\frac{\left\| \boldsymbol{X}_L^T \boldsymbol{\ell}_i^{(\gamma_2)} \right\|_2}{\left\| \boldsymbol{X}_L^T \boldsymbol{\ell}_i^{(\gamma_2)} \right\|_2 + \gamma_1 \left\| {\boldsymbol{X}_U^{(\gamma_2)}}^T \boldsymbol{u}_i^{(\gamma_2)} \right\|_2} \right) \hat{c}_i^{(\gamma_2)} \boldsymbol{w}_i^{(\gamma_2)}$$

is a generalization of Display (4) and balances the degree of coefficient shrinkage by the relative extents of U- versus L-extrapolations in the i^{th} direction as tuning parameter γ_1 is increased.

The Figure 2 block extrapolation example is now revisited with the notation of Display (18) and other mathematical developments from this section in mind. Extrapolation directions can always be computed with Proposition 5. When $p = 2$, the 1^{st} extrapolation direction is comprised of all vectors orthogonal to $\boldsymbol{w}_2^{(\gamma_2)}$, and the 2^{nd} extrapolation direction is comprised of all vectors orthogonal to $\boldsymbol{w}_1^{(\gamma_2)}$. Directions and extents in Figure 2 were all based on the semi-supervised extreme setting $\gamma_2 = \infty$. In this example, the extent of U-extrapolation is a larger multiple of the L-extent in the 1^{st} direction, so $\tau_1^{(\gamma_2)} > \tau_2^{(\gamma_2)}$ is a strict inequality. In addition, U-extents have the larger magnitude, so $\tau_2^{(\gamma_2)} > 1$ is another artifact of this particular example. An example of $p > 2$ is deferred until discussion of Figure 6 in the examples Section 4.5.

4.2 Joint Trained Ridge Regression

Estimator (9) with $\boldsymbol{\lambda} = (0, \lambda_2)$ is motivated with augmented labeled data

$$\boldsymbol{X}_L^{(\lambda_2)} = \begin{pmatrix} \boldsymbol{X}_L \\ \sqrt{\lambda_2}\boldsymbol{I} \end{pmatrix} \text{ and } \boldsymbol{Y}_L^{\star} = \begin{pmatrix} \boldsymbol{Y}_L \\ \vec{0} \end{pmatrix} \tag{21}$$

having p additional rows. The resulting *joint trained ridge estimator*

$$\widehat{\boldsymbol{\beta}}_{\boldsymbol{\gamma},(0,\lambda_2)} = \arg\min_{\boldsymbol{\beta}} \|\boldsymbol{Y}_L - \boldsymbol{X}_L\boldsymbol{\beta}\|_2^2 + \gamma_1 \left\| \boldsymbol{X}_U^{(\gamma_2)}\boldsymbol{\beta} \right\|_2^2 + \lambda_2 \|\boldsymbol{\beta}\|_2^2$$

is equivalent to Joint Trained Least Squares (11) given Data (21). Hence,

$$\widehat{\boldsymbol{\beta}}_{\boldsymbol{\gamma},(0,\lambda_2)} = \left({\boldsymbol{X}_L^{(\lambda_2)}}^T \boldsymbol{X}_L^{(\lambda_2)} + \gamma_1 {\boldsymbol{X}_U^{(\gamma_2)}}^T \boldsymbol{X}_U^{(\gamma_2)} \right)^{-1} \left({\boldsymbol{X}_L^{(\lambda_2)}}^T \boldsymbol{X}_L^{(\lambda_2)} \right) \widehat{\boldsymbol{\beta}}_{\lambda_2}^{(\text{RIDGE})}, \tag{22}$$

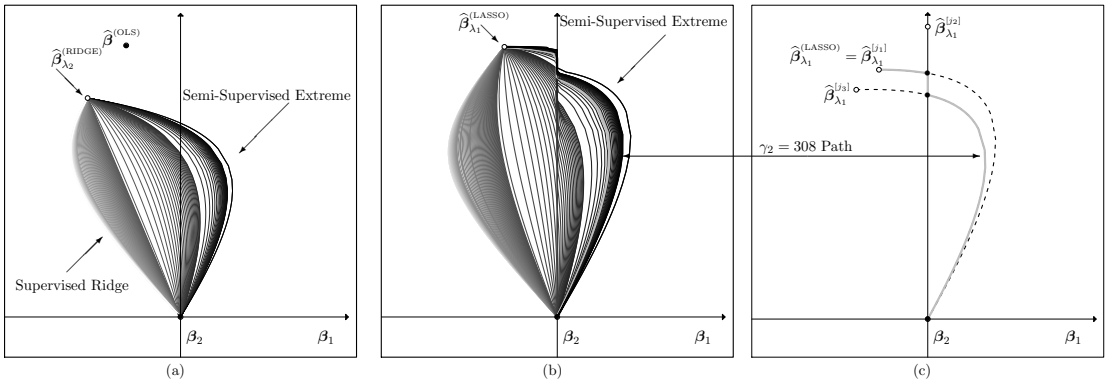


Figure 5: Paths of candidate $\widehat{\boldsymbol{\beta}}_{\boldsymbol{\gamma},\boldsymbol{\lambda}}$ for the Figure 1 block example varying $\gamma_1 > 0$ with darker curves for larger $\gamma_2 > 0$ are compared. (a) Joint trained ridge paths at a fixed $\boldsymbol{\lambda} = (0, 0.1)$ start at supervised ridge $\widehat{\boldsymbol{\beta}}^{(\text{RIDGE})}_{\lambda_2}$ instead of supervised OLS $\widehat{\boldsymbol{\beta}}^{(\text{OLS})}$. (b) Similarly, joint trained lasso paths at a fixed $\boldsymbol{\lambda} = (0.01, 0)$ start at supervised lasso $\widehat{\boldsymbol{\beta}}^{(\text{LASSO})}_{\lambda_1}$. However, these continuous paths are not differentiable at points where the active set changes. (c) The path from (b) with $\gamma_2 = 308$ is highlighted. Active set changes are marked by bullets $\bullet$, and the reference curves based on the right of Equation (23) are also displayed as dashed lines for $i = 1, 2, 3$. Each reference curve starts at a $\widehat{\boldsymbol{\beta}}^{[j_i]}_{\lambda_1}$ (marked by an open circle $\circ$) and terminates at the origin. The actual candidate path always equals one of the displayed reference curves. It starts at $\widehat{\boldsymbol{\beta}}^{(\text{LASSO})}_{\lambda_1} = \widehat{\boldsymbol{\beta}}^{[j_1]}_{\lambda_1}$ when $\gamma_1 = 0$ and switches reference curves whenever there is a change in the active set.

because $\widehat{\boldsymbol{\beta}}^{(\text{RIDGE})}_{\lambda_2} = \left({\boldsymbol{X}_L^{(\lambda_2)}}^T \boldsymbol{X}_L^{(\lambda_2)} \right)^{-1} \boldsymbol{X}_L^T \boldsymbol{Y}_L$ is the OLS estimator given Data (21). Matrix ${\boldsymbol{X}_L^{(\lambda_2)}}^T \boldsymbol{X}_L^{(\lambda_2)} = \boldsymbol{X}_L^T \boldsymbol{X}_L + \lambda_2 \boldsymbol{I}$ with $\lambda_2 > 0$ is positive definite, so the inverse required to compute $\widehat{\boldsymbol{\beta}}_{\boldsymbol{\gamma},(0,\lambda_2)}$ exists. Estimates (22) for the block extrapolation example come out as expected in Figure 5(a). Paths start at $\widehat{\boldsymbol{\beta}}^{(\text{RIDGE})}_{\lambda_2}$ with $\lambda_2 = 0.1$ and converge to the origin.

4.3 Joint Trained Lasso Regression

Supervised Optimization (7) with $\lambda_2 = 0$ simplifies to $\widehat{\boldsymbol{\beta}}^{(\text{LASSO})}_{\lambda_1} = \widehat{\boldsymbol{\beta}}^{(\text{ENET})}_{\lambda_1,0}$, a well-understood technique for incorporating variable selection when p is large and the columns of $\boldsymbol{X}_L$ are linearly independent (Friedman et al., 2010). The goal in this section is to use what is already known about $\widehat{\boldsymbol{\beta}}^{(\text{LASSO})}_{\lambda_1}$ to provide an understanding of the *joint trained lasso* $\widehat{\boldsymbol{\beta}}_{\boldsymbol{\gamma},(\lambda_1,0)}$ from Problem (9). Denote the *active set* of some estimate $\widehat{\boldsymbol{\beta}}$ by $\mathcal{A} \subset \{1, \ldots, p\}$, so $\left(\widehat{\boldsymbol{\beta}}\right)_{\mathcal{A}}$ is its $|\mathcal{A}| \times 1$ vector of nonzero components and $\left(\widehat{\boldsymbol{\beta}}\right)_{\bar{\mathcal{A}}} = \vec{0}$ is $(p - |\mathcal{A}|) \times 1$. Also denote its *sign vector* by $\boldsymbol{s} = \text{sign}\left(\left(\widehat{\boldsymbol{\beta}}\right)_{\mathcal{A}}\right)$ and the $|L| \times |\mathcal{A}|$ sub matrix of $\boldsymbol{X}$ with labeled rows and

active set columns by $\boldsymbol{X}_{L\mathcal{A}}$. The active set $\mathcal{A}^{(\text{SUP})}$ and sign vector $\boldsymbol{s}^{(\text{SUP})}$ of the supervised lasso at a given λ_1 satisfy the constraint

$$\boldsymbol{X}^T_{L\mathcal{A}^{(\text{SUP})}}\boldsymbol{X}_{L\mathcal{A}^{(\text{SUP})}}\left(\widehat{\boldsymbol{\beta}}^{(\text{LASSO})}_{\lambda_1}\right)_{\mathcal{A}^{(\text{SUP})}} = \boldsymbol{X}^T_{L\mathcal{A}^{(\text{SUP})}}\boldsymbol{Y}_L - \lambda_1\boldsymbol{s}^{(\text{SUP})}.$$

Estimates $\widehat{\boldsymbol{\beta}}^{(\text{LASSO})}_{\lambda_1}$ are a differentiable function in λ_1 with a finite number of exceptions. This function is continuous, but not differentiable when the active set changes.

The joint trained lasso $\widehat{\boldsymbol{\beta}}_{\boldsymbol{\gamma},(\lambda_1,0)}$ has properties similar to the supervised lasso by Optimization (9), because it's a lasso estimator with unlabeled imputations $\boldsymbol{Y}_U = \vec{0}$ and modified $\boldsymbol{X}$. Unlike joint trained ridge and joint trained least squares from Sections 4.1 and 4.2, the joint trained lasso coefficients are not always a linear combination of the supervised lasso, and this complicates its ensuing interpretation. There are $2^p + 2p + 1$ active-set/sign-vector combinations for any $p \geq 2$. For example, when $p = 2$, there are nine combinations, i.e., $2^2 = 4$ quadrants, $2 \times 2 = 4$ axial directions, and 1 origin. Each active-set/sign-vector combination has a set of reference coefficients $\left(\widehat{\boldsymbol{\beta}}^{[j]}_{\lambda_1}\right)_{\mathcal{A}_j} = \left(\boldsymbol{X}^T_{L\mathcal{A}_j}\boldsymbol{X}_{L\mathcal{A}_j}\right)^{-1}\left(\boldsymbol{X}^T_{L\mathcal{A}_j}\boldsymbol{Y}_L - \lambda_1\boldsymbol{s}_j\right)$ and $\left(\widehat{\boldsymbol{\beta}}^{[j]}_{\lambda_1}\right)_{\bar{\mathcal{A}}_j} = \vec{0}$ for $j = 1, \ldots, 2^p + 2p + 1$. These reference coefficients have important properties. First, $\widehat{\boldsymbol{\beta}}^{[j]}_{\lambda_1}$ are independent of $\boldsymbol{X}_U$. Second, there exists a $j \in \{1, \ldots, 2^p + 2p + 1\}$ such that $\widehat{\boldsymbol{\beta}}^{(\text{LASSO})}_{\lambda_1} = \widehat{\boldsymbol{\beta}}^{[j]}_{\lambda_1}$. Third, $\text{sign}\left(\left(\widehat{\boldsymbol{\beta}}^{[j]}_{\lambda_1}\right)_{\mathcal{A}_j}\right)$ does not necessarily equal $\boldsymbol{s}_j$. Next, the path of the joint trained lasso as a function of γ_1 at a given γ_2 is studied. Let the finite set $\{a_i\}_{i=1}^k$ be the finite values of γ_1 where the active set of the joint trained lasso changes and define $a_0 = 0$ and $a_{k+1} = \infty$. Also define the subsequence $j_1, \ldots, j_k$ such that $\mathcal{A}_{j_i}$ and $\boldsymbol{s}_{j_i}$ correspond to the joint trained lasso for any $\gamma_1 \in [a_{i-1}, a_i)$, so this subsequence tracks the evolution of the joint trained lasso's active set and sign vector. Thus, for all $\gamma_1 \in [a_{i-1}, a_i)$,

$$\left(\widehat{\boldsymbol{\beta}}_{\boldsymbol{\gamma},(\lambda_1,0)}\right)_{\mathcal{A}_{j_i}} = \left(\boldsymbol{X}^T_{L\mathcal{A}_{j_i}}\boldsymbol{X}_{L\mathcal{A}_{j_i}} + \gamma_1\boldsymbol{X}^{(\gamma_2)^T}_{U\mathcal{A}_{j_i}}\boldsymbol{X}^{(\gamma_2)}_{U\mathcal{A}_{j_i}}\right)^{-1}\boldsymbol{X}^T_{L\mathcal{A}_{j_i}}\boldsymbol{X}_{L\mathcal{A}_{j_i}}\left(\widehat{\boldsymbol{\beta}}^{[j_i]}_{\lambda_1}\right)_{\mathcal{A}_{j_i}}, \tag{23}$$

and shrinking of regression coefficients on the active set looks very much like Display (14).

i	1	2	3	4
$\mathcal{A}_{j_i}$	$\{1,2\}$	$\{2\}$	$\{1,2\}$	$\emptyset$
$\boldsymbol{s}^T_{j_i}$	$(-1,1)$	$(0,1)$	$(1,1)$	-
γ_1	$[0, 0.004)$	$[0.004, 0.008)$	$[0.008, \infty)$	∞

Table 1: Block extrapolation active-set, sign-vector combinations are listed as a function of γ_1 for the joint trained lasso coefficients $\widehat{\boldsymbol{\beta}}_{\boldsymbol{\gamma},\boldsymbol{\lambda}}$ from Figure 5(c) with $\boldsymbol{\lambda} = (0.01, 0)$ and $\gamma_2 = 308$.

Figure 5(b) plots paths of vectors $\widehat{\boldsymbol{\beta}}_{\boldsymbol{\gamma},(\lambda_1,0)}$ by γ_2 as a function of γ_1 at $\lambda_1 = 0.01$ for the block extrapolation example. The semi-supervised path starts at the supervised estimate $\widehat{\boldsymbol{\beta}}^{(\text{LASSO})}_{\lambda_1}$ when $\gamma_1 = 0$. Equation (23) establishes a local property of the joint trained lasso. The approach has the same active set and sign vector as the supervised coefficient for a

small region $\gamma_1 \in [0, a_1)$, where $a_1 > 0$. This local property of the joint trained lasso, which was mathematically verified in this section, is stated as a key assumption while deriving the general performance bounds in Section 5. An example is the highlighted path with $\gamma_2 = 308$ from Figure 5(b) shown in Figure 5(c). This candidate path of semi-supervised regression coefficients visits four active-set, sign-vector combinations as a continuous function of γ_1 at given λ_1 and γ_2. These visited combinations are listed in Table 1 along with their corresponding values γ_1. Figure 5(c) also includes dashed reference curves based on the right of Equation (23) as a function of γ_1 for each non-empty active-set/sign-vector combination visited by the approach, i.e., $i = 1, 2, 3$. The candidate semi-supervised estimates follow along a reference path until the active set changes, and then the path switches to the reference path with the new active set and sign vector. This continues until the path terminates at the origin when $\gamma_1 = \infty$.

4.4 Joint Trained Elastic Net Regression

A general view of Problem (9) when all four tuning parameters are finite and positive comes from stringing concepts from Sections 4.2 and 4.3 together. In particular,

$$\left(\widehat{\boldsymbol{\beta}}_{\boldsymbol{\gamma},\boldsymbol{\lambda}}\right)_{\mathcal{A}_{j_i}} = \left(\boldsymbol{X}_{L\mathcal{A}_{j_i}}^{(\lambda_2)\ T}\boldsymbol{X}_{L\mathcal{A}_{j_i}}^{(\lambda_2)} + \gamma_1 \boldsymbol{X}_{U\mathcal{A}_{j_i}}^{(\gamma_2)\ T}\boldsymbol{X}_{U\mathcal{A}_{j_i}}^{(\gamma_2)}\right)^{-1}\left(\boldsymbol{X}_{L\mathcal{A}_{j_i}}^{(\lambda_2)\ T}\boldsymbol{X}_{L\mathcal{A}_{j_i}}^{(\lambda_2)}\right)\left(\widehat{\boldsymbol{\beta}}_{\boldsymbol{\lambda}}^{[j_i]}\right)_{\mathcal{A}_{j_i}},$$

where $\mathcal{A}_{j_i}$ and $\boldsymbol{s}_{j_i}$ depend on $(\boldsymbol{\gamma}, \boldsymbol{\lambda})$ and

$$\left(\widehat{\boldsymbol{\beta}}_{\boldsymbol{\lambda}}^{[j_i]}\right)_{\mathcal{A}_{j_i}} = (1+\lambda_2)\left(\boldsymbol{X}_{L\mathcal{A}_{j_i}}^{(\lambda_2)\ T}\boldsymbol{X}_{L\mathcal{A}_{j_i}}^{(\lambda_2)}\right)^{-1}\left(\boldsymbol{X}_{L\mathcal{A}_{j_i}}^{(\lambda_2)^T}\boldsymbol{Y}_L - \lambda_1 \boldsymbol{s}_{j_i}\right).$$

The order of operations are important: substitute $\boldsymbol{X}_{L\mathcal{A}_{j_i}}$ for $\boldsymbol{X}_L$ and then apply Equation (21) to get $\boldsymbol{X}_{L\mathcal{A}_{j_i}}^{(\lambda_2)}$, and similarly, $\boldsymbol{X}_{U\mathcal{A}_{j_i}}$ for $\boldsymbol{X}_U$ to then get $\boldsymbol{X}_{U\mathcal{A}_{j_i}}^{(\gamma_2)}$ from Equation (8). Increased γ_1 and γ_2 puts more emphasis on shrinking unlabeled fits. Increased λ_2 and/or decreased γ_2 results in the labeled and/or unlabeled directions being better approximated by an $|\mathcal{A}_{j_i}|$-sphere, and increased λ_1 for presumably more stringent variable selection. Cross-validation often selects the joint trained elastic net with strictly positive lasso $\lambda_1 > 0$ and ridge $\lambda_2 > 0$ tuning parameter values in practical applications, so the joint trained elastic net is showcased later through its performance on numerical examples (i.e., simulated and real data sets) in Section 6.

4.5 Geometric Extrapolation Examples

The purpose of this section is learn more about the properties of our semi-supervised adjustment through additional geometrical examples of joint trained least squares from Section 4.1. Recall the joint trained least squares example in Figures 1, 2, and 4 for the heavily studied block extrapolation example. The first row of Figure 6 motivates additional discussion by simply changing the unlabeled feature data as follows.

- "Pure" – Extrapolations of larger magnitude are roughly in-line with the 2$^{\text{nd}}$ principal component, so supervised and semi-supervised shrinking are in similar directions at varying degrees.

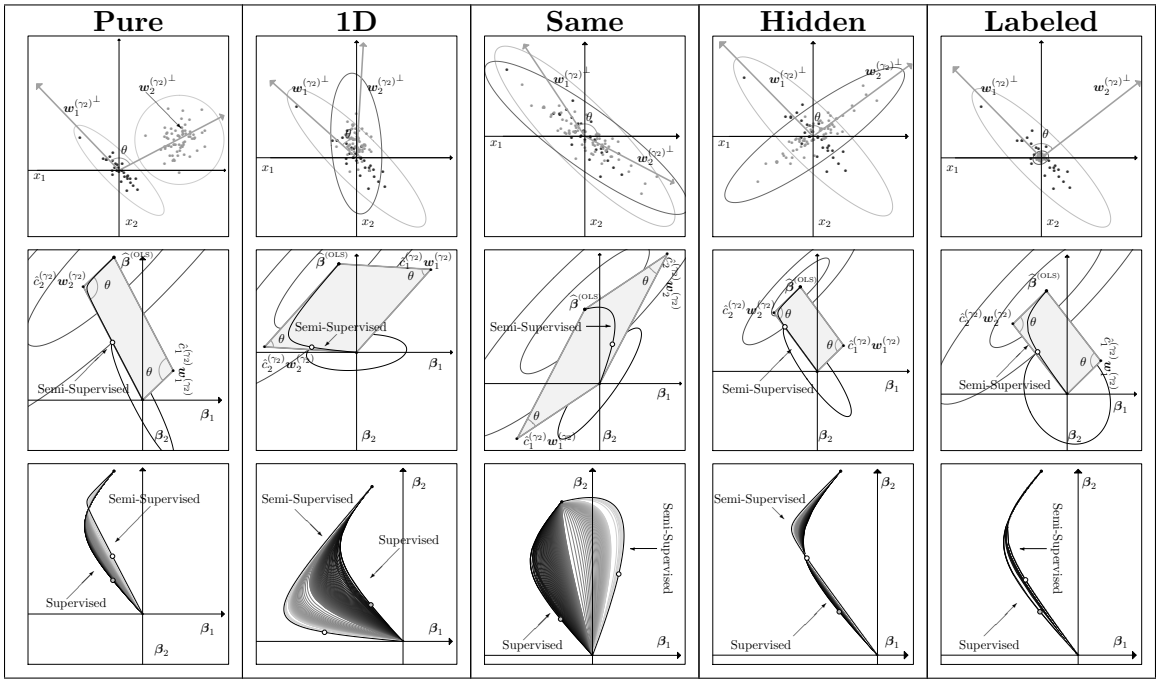


Figure 6: An additional geometrical example of joint trained least squares is displayed in each column. Row 1: Only the unlabeled feature data $\boldsymbol{X}_U$ from the "working" block extrapolation example from Figures 1, 2, and 4 were changed. Row 2: Ellipses (12) and (13) intersect at a point on the semi-supervised extreme. Row 3: Paths $\widehat{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}$ are plotted by γ_2 varying γ_1. The gray circle is the supervised ridge solution from Figure 4(a).

- "1D" – The unlabeled marginal distribution is more volatile in one dimension x_2.
- "Same" – Minor discrepancies arise naturally in empirical distributions when taking independent samples from the same distribution.
- "Hidden" – Components x_1 and x_2 have roughly the same marginal distributions in both sets, but unlabeled extrapolations are hidden in the bivariate distribution of (x_1, x_2).
- "Labeled" – Only the labeled feature data deviate substantially from the origin.

Broader sets of candidate $\widehat{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}$ are entertained in the block, 1D, and same extrapolation examples. On the other hand, directions of extrapolations are roughly the principal components in the pure, hidden, and labeled extrapolation examples, and these examples have smaller candidates sets $\widehat{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}$ as a result. In general, such smaller candidates sets are expected whenever the semi-supervised eigenvector directions of shrinking based on $\left(\boldsymbol{X}_L^T \boldsymbol{X}_L\right)^{-1} {\boldsymbol{X}_U^{(\gamma_2)}}^T \boldsymbol{X}_U^{(\gamma_2)}$ are approximately those in supervised ridge regression based on

$\tau_i^{(\gamma_2)}$	Block	Pure	1D	Same	Hidden	Labeled
$\tau_1^{(\gamma_2)}$	95.1	662.5	11.2	4.2	38.0	0.30
$\tau_2^{(\gamma_2)}$	22.6	1.3	0.3	1.2	0.1	0.01

Table 2: The eigenvalues of $\boldsymbol{M}^{(\gamma_2)}$ with $\gamma_2 = \infty$ are listed.

$\boldsymbol{X}_L^T \boldsymbol{X}_L$, but this does not imply that supervised and semi-supervised ridge techniques are approximately the same (see Remark 8).

The block and pure examples emphasize profoundly different directions of extrapolation, but have eigenvalues of large magnitude in Table 2. Extrapolations are on separate manifolds, and the approach shrinks predictions much more in these two examples at a given $\gamma_1 > 0$, by Equation (16). The semi-supervised extreme path closely maps the sides of its bounding parallelogram from Proposition 4 in the pure and hidden examples because their $\tau_i^{(\gamma_2)}$ in Table 2 are of different orders of magnitude. This phenomena is not present in the block and same examples when eigenvalues are of the same order of magnitude. The semi-supervised extreme in the 1D example is of special note. Its labeled feature data are negatively correlated, so the extreme emphasizes x_1 to shrink the influence of the component x_2 which is volatile in the unlabeled data.

Figure 7 is a 3D example. In the semi-supervised extreme, the shrinking matrix $\boldsymbol{M}^{(\gamma_2)}$ has eigenvalues $\tau_i^{(\gamma_2)} = 2090, 21.3, 1.08$, so shrinking of regression coefficients is much more heavily focused in direction $\boldsymbol{w}_1^{(\gamma_2)}$ because these eigenvalues differ in magnitude. The 1st direction of extrapolation is based on the other $p - 1 = 2$ directions of coefficient shrinking $\boldsymbol{w}_2^{(\gamma_2)}$ and $\boldsymbol{w}_3^{(\gamma_2)}$ and is defined as the set of all feature vectors that are orthogonal to both of these directions. The desired effect of using the unlabeled data to shrink unlabeled extrapolations more is achieved through Equation (16) at any $\gamma_1 > 0$. Semi-supervised predictions are $\boldsymbol{x}_0^T \widehat{\boldsymbol{\beta}}^{(\text{OLS})}/(1 + \gamma_1 2090)$ if $\boldsymbol{x}_0$ is a feature vector on the 1st direction of extrapolation; $\boldsymbol{x}_0^T \widehat{\boldsymbol{\beta}}^{(\text{OLS})}/(1 + \gamma_1 21.3)$ if $\boldsymbol{x}_0$ is on the 2nd direction; and $\boldsymbol{x}_0^T \widehat{\boldsymbol{\beta}}^{(\text{OLS})}/(1 + \gamma_1 1.08)$ if $\boldsymbol{x}_0$ is on the 3rd direction. Candidate vectors $\widehat{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}$ in Figure 7(b) form a curved surface between supervised and semi-supervised extreme.

Remark 8 *Even if supervised and semi-supervised candidate sets $\widehat{\boldsymbol{\beta}}$ are approximately equal, semi-supervised training with the unlabeled feature data $\boldsymbol{X}_U$ may pick a very different (and hopefully more advantageous) estimate $\widehat{\boldsymbol{\beta}}$ within the candidate set during cross-validation. In general, whether or not such apparent "parameter redundancies" exist, we always advocate the use of supervised regularization ($\boldsymbol{\lambda} \neq \vec{0}$) together with semi-supervised regularization ($\boldsymbol{\gamma} \neq \vec{0}$), especially when p is large. Many parameter redundancies noted in the $p = 2$ examples are not present in large p applications. If one briefly backs up to the case of $p = 1$, all candidate paths from Section 4.1 essentially start on the number line at the OLS estimate and then shrink to zero. When $p = 3$, one could overlay $\widehat{\boldsymbol{\beta}}_{\boldsymbol{\gamma},(0,\lambda_2)}$ for all $\boldsymbol{\gamma} \in [0, \infty]^2$ at fixed $\lambda_2 > 0$, and this in-fact adds a distinct layer to the 3D surface in Figure 7(b). The key point is to broaden the choices in an intelligent manner as needed so that a most desirable $\widehat{\boldsymbol{\beta}}$ can be selected for the purpose of unlabeled prediction.*

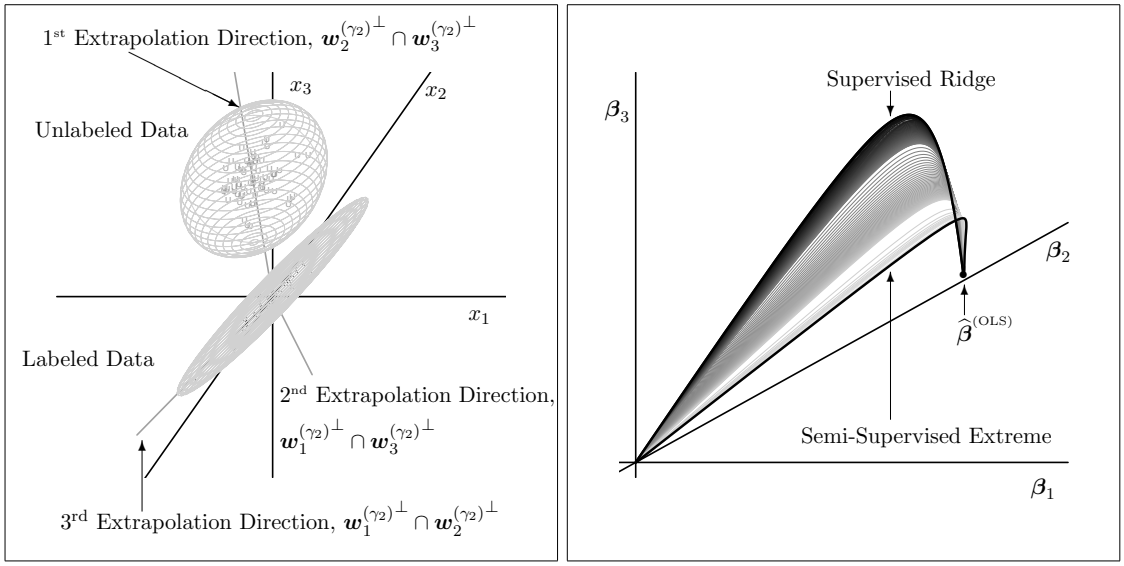


Figure 7: A $p = 3$ extrapolation data set is displayed. (a) The feature data along with the three extrapolation directions in the semi-supervised extreme of $\gamma_2 = \infty$ are plotted. Each direction of extrapolation is a line that equals the intersection of two planes by Proposition 5. (b) Candidate paths $\widehat{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}$ by γ_2 varying γ_1 have a nonlinear compromise between supervised ridge and the semi-supervised extreme.

5. Performance Bounds

A general sufficient condition is given in this section for when a semi-supervised adjustment improves expected unlabeled prediction performance for a large class of linear supervised approaches. Assumption 1 on the class of supervised approaches is a necessary but not a sufficient condition for the elastic net; this generality was intentional. Assumption 2 characterizes a local property of our semi-supervised adjustment that follows from its Section 4 geometry.

Assumption 1: *The supervised estimate $\widehat{\boldsymbol{\beta}}_{\boldsymbol{\lambda}}^{(SUP)}$ is unique for data $(\boldsymbol{X}_L, \boldsymbol{Y}_L)$ and some $\boldsymbol{\lambda}$. Let $\boldsymbol{\phi} = \{\boldsymbol{\lambda}, \mathcal{A}, \boldsymbol{s}\}$ and $q = |\mathcal{A}|$ denote its fixed properties.*

Assumption 2: *$\exists\ \delta > 0$ such that $\forall\ \gamma_1 \in [0, \delta)$ semi-supervised estimates $\widehat{\boldsymbol{\beta}}_{\gamma_1}^{(\boldsymbol{\phi})}$ have the supervised active set $\mathcal{A}$ and sign vector $\boldsymbol{s}$, and*

$$\begin{aligned} \left(\widehat{\boldsymbol{\beta}}_{\gamma_1}^{(\boldsymbol{\phi})}\right)_{\mathcal{A}} &= \left(\boldsymbol{I} + \gamma_1 \boldsymbol{M}_{\mathcal{A}}^{(\lambda_2,\infty)}\right)^{-1} \left(\widehat{\boldsymbol{\beta}}_{\boldsymbol{\lambda}}^{(SUP)}\right)_{\mathcal{A}}, \text{ where} \\ \boldsymbol{M}_{\mathcal{A}}^{(\lambda_2,\gamma_2)} &= \left(\boldsymbol{X}_{L\mathcal{A}}^{(\lambda_2)^T} \boldsymbol{X}_{L\mathcal{A}}^{(\lambda_2)}\right)^{-1} \boldsymbol{X}_{U\mathcal{A}}^{(\gamma_2)^T} \boldsymbol{X}_{U\mathcal{A}}^{(\gamma_2)}. \end{aligned}$$

Assumptions 1 and 2 always hold for the Joint Trained Optimization Problem (6) when $\lambda_2 > 0$ or $\mathrm{rank}(\boldsymbol{X}_L) = p$. For example, consider the joint trained lasso example from Figure 5(b) and Table 1. Assumption 1 holds with $\boldsymbol{\phi} = \{(0.01, 0), \{1, 2\}, (-1, 1)\}$, and Assumption 2 holds with $\delta = 0.004$ from Table 1.

Results to come focus on the impact of semi-supervised learning with $\gamma_2 = \infty$, so Propositions 3-6 are applied to $\widehat{\boldsymbol{\beta}}_{\gamma_1}^{(\boldsymbol{\phi})}$ and $\boldsymbol{M}_{\mathcal{A}}^{(\lambda_2,\infty)}$. Let $\left\{\left(\boldsymbol{w}_i^{(\boldsymbol{\phi})}, \tau_i^{(\boldsymbol{\phi})}\right)\right\}_{i=1}^{q}$ be an eigenbasis of $\boldsymbol{M}_{\mathcal{A}}^{(\lambda_2,\infty)}$ such that $\left\|\boldsymbol{X}_{L\mathcal{A}}^{(\lambda_2)}\boldsymbol{w}_i^{(\boldsymbol{\phi})}\right\|_2^2 = 1$ and $\left(\widehat{\boldsymbol{\beta}}_{\boldsymbol{\lambda}}^{(\text{SUP})}\right)_{\mathcal{A}} = \sum_{i=1}^{q}\hat{c}_i^{(\boldsymbol{\phi})}\boldsymbol{w}_i^{(\boldsymbol{\phi})}$ generalize Equation (15). Assumption 2 implies that Equation (16) generalizes to

$$\left(\widehat{\boldsymbol{\beta}}_{\gamma_1}^{(\boldsymbol{\phi})}\right)_{\mathcal{A}} = \left(\tfrac{1}{1+\gamma_1\tau_1^{(\boldsymbol{\phi})}}\right)\hat{c}_1^{(\boldsymbol{\phi})}\boldsymbol{w}_1^{(\boldsymbol{\phi})} + \cdots + \left(\tfrac{1}{1+\gamma_1\tau_q^{(\boldsymbol{\phi})}}\right)\hat{c}_q^{(\boldsymbol{\phi})}\boldsymbol{w}_q^{(\boldsymbol{\phi})}. \tag{24}$$

Assume the linear model with $\mathbb{E}[\boldsymbol{Y}] = \boldsymbol{X}\boldsymbol{\beta}$ and $\mathbb{V}\text{ar}(\boldsymbol{Y}) = \sigma^2\boldsymbol{I}$ and project

$$\boldsymbol{\beta}_{\mathcal{A}} = c_1^{(\boldsymbol{\phi})}\boldsymbol{w}_1^{(\boldsymbol{\phi})} + \cdots + c_q^{(\boldsymbol{\phi})}\boldsymbol{w}_q^{(\boldsymbol{\phi})}. \tag{25}$$

If small $c_i^{(\boldsymbol{\phi})}$ correspond to large $\tau_i^{(\boldsymbol{\phi})}$, a performance improvement on the evaluation function $\left\|\boldsymbol{X}_{U\mathcal{A}}\left(\boldsymbol{\beta}_{\mathcal{A}} - \left(\widehat{\boldsymbol{\beta}}_{\gamma_1}^{(\boldsymbol{\phi})}\right)_{\mathcal{A}}\right)\right\|_2^2$ appears likely by Equations (24) and (25). If $\tau_i^{(\boldsymbol{\phi})}$ is large for a small subset $i \in \Omega \subset \{1,\ldots,p\}$ and small otherwise, then semi-supervised performance is expected to be better over a larger percentage of the possible directions for the true $\boldsymbol{\beta}$. Such high performance circumstances occur when a low dimensional manifold of $\boldsymbol{X}_{U\mathcal{A}}$ concentrates away from that of $\boldsymbol{X}_{L\mathcal{A}}$ and the true coefficient vector $\boldsymbol{\beta}$ emphasizes directions dominated by labeled extrapolations. Assumption 3 helps establish a general transductive bound for when semi-supervised learning is better than supervised on evaluation function $\mathbb{E}\left[\left\|\boldsymbol{X}_U\left(\widehat{\boldsymbol{\beta}}_{\gamma_1}^{(\boldsymbol{\phi})} - \boldsymbol{\beta}\right)\right\|_2^2\middle|\boldsymbol{\phi}\right]$.

Assumption 3: $\mathbb{E}\left[\hat{c}_i^{(\boldsymbol{\phi})}\middle|\boldsymbol{\phi}\right] = \mu_i < \infty$ *and* $\mathbb{V}\text{ar}\left[\hat{c}_i^{(\boldsymbol{\phi})}\middle|\boldsymbol{\phi}\right] = \sigma_i^2 < \infty\ \forall i \in \{1,\ldots,q\}$.

Let $\bar{\mathcal{A}} = \{1,\ldots,p\} - \mathcal{A}$ be the supervised non-active set and define $\boldsymbol{X}_{U\emptyset}\boldsymbol{\beta}_{\emptyset} = \vec{0}$. Theorem 9 provides a sufficient condition on parameters $\left(\boldsymbol{\beta}, \sigma^2\right)$ for when semi-supervised outperforms supervised given the feature data and $\boldsymbol{\phi}$.

Theorem 9 *Let Assumptions 1-3 hold. Also, let* $q \geq 1$, $\tau_1^{(\boldsymbol{\phi})} > 0$, *and* $p_i\left(\boldsymbol{\tau}^{(\boldsymbol{\phi})}\right) = \frac{\tau_i^{(\boldsymbol{\phi})^2}\sigma_i^2}{\sum_{j=1}^{q}\tau_j^{(\boldsymbol{\phi})^2}\sigma_j^2}$. *If* $\sum_{i=1}^{q} p_i\left(\boldsymbol{\tau}^{(\boldsymbol{\phi})}\right)\left(\frac{\mu_i\left(c_i^{(\boldsymbol{\phi})} + \boldsymbol{u}_i^{(\boldsymbol{\phi})^T}\boldsymbol{X}_{U\bar{\mathcal{A}}}\boldsymbol{\beta}_{\bar{\mathcal{A}}}/\sqrt{\kappa_i^{(\gamma_2)}}\right) - \mu_i^2}{\sigma_i^2}\right) < 1$, *then*

$$\mathbb{E}\left[\left\|\boldsymbol{X}_U\left(\widehat{\boldsymbol{\beta}}_{\gamma_1}^{(\boldsymbol{\phi})} - \boldsymbol{\beta}\right)\right\|_2^2\middle|\boldsymbol{\phi}\right] < \mathbb{E}\left[\left\|\boldsymbol{X}_U\left(\widehat{\boldsymbol{\beta}}_{\boldsymbol{\lambda}}^{(SUP)} - \boldsymbol{\beta}\right)\right\|_2^2\middle|\boldsymbol{\phi}\right].$$

As stated earlier, Assumptions 1 and 2 hold for the general $\boldsymbol{\lambda}$ joint trained elastic net regression of Section 4.4. In the case of $\boldsymbol{\lambda} = \vec{0}$ least squares, it is also easily verified that $\mu_i = c_i^{(\boldsymbol{\phi})}$ and $\sigma_i^2 = \sigma^2$ for Assumption 3. The mathematical form of the extreme version of joint trained least squares in Equation (14) is equivalent to that for generalized ridge regression. Corollary 10 in conjunction with Casella (1980) shows that joint trained least squares is

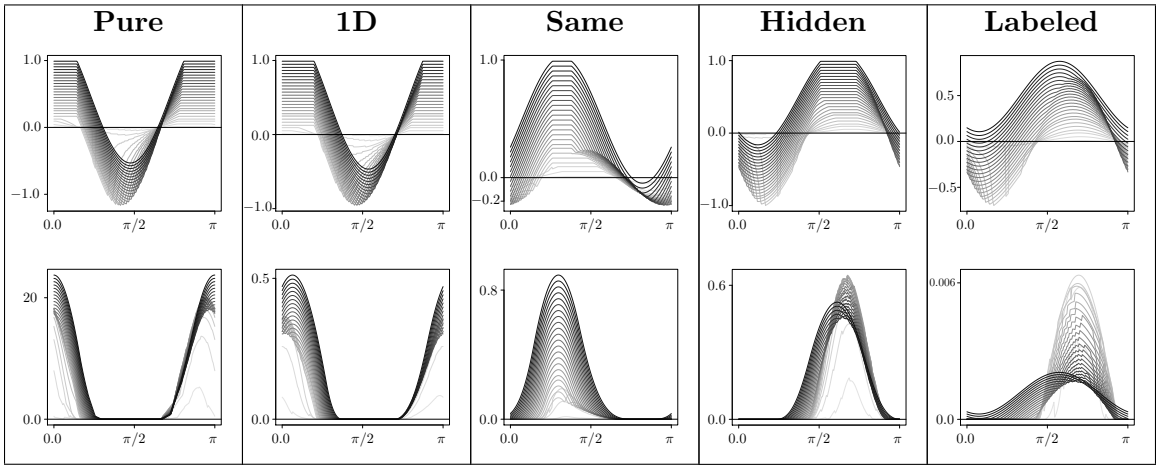


Figure 8: The five examples with $p = 2$ from Figure 6 are revisited. Row 1: Theoretical bound $\sigma^2 - \sigma^2_{\text{LB}}(\boldsymbol{\beta}(\vartheta))$ is plotted against ϑ. Darker curves correspond to larger $\sigma^2 \in [0, 1]$. Row 2: The corresponding differences $\text{RMSE}_U^{(\text{SUP})} - \text{RMSE}_U^{(\text{SEMI})}$ are plotted against ϑ.

asymptotically minimax with respect to loss function $\mathbb{E}\left[\left\|\boldsymbol{X}_U\left(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}\right)\right\|_2^2\right]$ as $|L| \to \infty$. In the case of ridge regression, $\mu_i = c_i^{(\boldsymbol{\phi})} - \lambda_2 \boldsymbol{w}_i^{(\boldsymbol{\phi})^T}\boldsymbol{\beta}$ and $\sigma_i^2 = \boldsymbol{w}_i^{(\boldsymbol{\phi})^T}\boldsymbol{X}_L^T\boldsymbol{X}_L\boldsymbol{w}_i^{(\boldsymbol{\phi})}\sigma^2$ for any $i \in \{1, \ldots, p\}$ are also straightforward to derive, so Theorem 9 reduces to Corollary 11.

Corollary 10 *Joint trained least squares with $\gamma_2 = \infty$ dominates supervised least squares in prediction on $\boldsymbol{X}_U$ if $q \geq 1$ and $\tau_1^{(\boldsymbol{\phi})} > 0$.*

Corollary 11 *The extreme version of joint trained ridge regression dominates supervised ridge regression in prediction on $\boldsymbol{X}_U$ if $q \geq 1$, $\tau_1^{(\boldsymbol{\phi})} > 0$, and*

$$\sigma^2_{LB}(\boldsymbol{\beta}) = \left(\sum_{i=1}^{p} p_i\left(\boldsymbol{\tau}^{(\boldsymbol{\phi})}\right)\left(\frac{\left(c_i^{(\boldsymbol{\phi})} - \lambda_2 \boldsymbol{w}_i^{(\boldsymbol{\phi})^T}\boldsymbol{\beta}\right)\left(\lambda_2 \boldsymbol{w}_i^{(\boldsymbol{\phi})^T}\boldsymbol{\beta}\right)}{\boldsymbol{w}_i^{(\boldsymbol{\phi})^T}\boldsymbol{X}_L^T\boldsymbol{X}_L\boldsymbol{w}_i^{(\boldsymbol{\phi})}}\right)\right)_+ < \sigma^2.$$

The block feature data from Figure 1 were used to construct Figure 3 and introduce the reader to the semi-supervised ridge bound $\sigma^2_{\text{LB}}(\boldsymbol{\beta})$ earlier in Section 2.2. The analog of that figure for the five examples from Figure 6 is given in this section by Figure 8. A technical explanation of how these figures were constructed precedes the qualitative discussion of their interpretations in the next paragraph. First, note that $\sigma^2_{\text{LB}}(\boldsymbol{\beta}(\vartheta))$ from Corollary 11 is independent of σ^2. It was computed for all $\boldsymbol{\beta}(\vartheta) = (\sin(\vartheta), \cos(\vartheta))^T$ over a fine grid of $\vartheta \in [0, \pi]$, and the $\sigma^2_{\text{LB}}(\boldsymbol{\beta}(\vartheta))$ were compared to a fine, equally spaced grid of $\sigma^2 \in [0, 1]$. Only the right half of the unit circle was considered for $\boldsymbol{\beta}$ because $\sigma^2_{\text{LB}}(\boldsymbol{\beta}(\vartheta)) = \sigma^2_{\text{LB}}(\boldsymbol{\beta}(\vartheta + \pi))$. Also, $\sigma^2_{\text{LB}}(r\boldsymbol{\beta}(\vartheta)) = r^2\sigma^2_{\text{LB}}(\boldsymbol{\beta}(\vartheta))$, so the same trend results from the scaled parameters $r\boldsymbol{\beta}(\vartheta)$ with $\sigma^2 \in [0, r^2]$. The ridge parameter was set to the "best" supervised attempt

of $\lambda_2^{(\text{opt})}$ minimizing $\mathbb{E}\left[\left\|\boldsymbol{X}_L\left(\boldsymbol{\beta}(\vartheta)-\widehat{\boldsymbol{\beta}}_{\lambda_2}^{(\text{RIDGE})}\right)\right\|_2^2\right]$. Interest was in identifying ϑ's when a semi-supervised adjustment helps, i.e., when $\sigma^2-\sigma^2_{\text{LB}}\left(\boldsymbol{\beta}(\vartheta)\right)>0$.

Angles ϑ corresponding to lucky $\boldsymbol{\beta}(\vartheta)$ and to reductions in RMSE due to semi-supervised learning line-up vertically across the rows of Figure 8 (i.e., ϑ with a positive vertical coordinate in row 1 also have a positive coordinate in row 2 and vice versa). Row 2 is the magnitude of the improvements, and the examples with the largest magnitude (i.e., the pure example and the block example from Figure 3(b)) are those with the largest eigenvalues in Table 2 as expected. The labeled example with the smallest improvements also has the smallest eigenvalues. Direction $\boldsymbol{w}_2^{(\boldsymbol{\phi})}$ (eyeballed from the row 1 of Figure 6) should be compared to row 1 of Figure 8. In each example, the center for potentially large improvements is roughly $\boldsymbol{\beta}(\vartheta)\propto\boldsymbol{w}_2^{(\boldsymbol{\phi})}$, and the center for little to no potential improvement is roughly $\boldsymbol{\beta}(\vartheta)\propto\boldsymbol{w}_2^{(\boldsymbol{\phi})^\perp}$. The generalization to $p\geq 2$ in Proposition 12 below extends this interpretation to that given back in Section 2.2. That is, if $\boldsymbol{\beta}$ is orthogonal to an unlabeled manifold, then $\widehat{\boldsymbol{\beta}}_{\gamma_1}^{(\boldsymbol{\phi})}$ has an unlabeled prediction advantage over $\widehat{\boldsymbol{\beta}}_{\lambda_2}^{(\text{RIDGE})}$, whereas $\boldsymbol{\beta}$ parallel to the unlabeled manifold yields no theoretical advantage.

Proposition 12 *If $\tau_1^{(\boldsymbol{\phi})}>0$ and $\boldsymbol{\beta}_i\in\bigcap_{j\in\{1,\ldots,p\}-\{i\}}\boldsymbol{w}_j^{(\boldsymbol{\phi})^\perp}$ is unit length, then the joint trained ridge performance bound from Corollary 11 satisfies $\sigma^2_{LB}\left(\boldsymbol{\beta}_i\right)\geq\lambda_2 p_i\left(\boldsymbol{\tau}^{(\boldsymbol{\phi})}\right)$ for $i\in\{1,\ldots,p\}$ and $\sigma^2_{LB}\left(\boldsymbol{\beta}_i\right)\geq\sigma^2_{LB}\left(\boldsymbol{w}_j^{(\boldsymbol{\phi})}/\left\|\boldsymbol{w}_j^{(\boldsymbol{\phi})}\right\|_2\right)$ if $j\geq i$.*

Given a lasso estimate $\widehat{\boldsymbol{\beta}}_{\lambda_1}^{(\text{LASSO})}$, response $\boldsymbol{Y}_L\in\mathcal{Y}_L(\boldsymbol{\phi})=\left\{\boldsymbol{y}\in I\!R^{|L|}:\widehat{\boldsymbol{\beta}}_{\lambda_1}^{(\text{LASSO})}\text{ has }\boldsymbol{\phi}\right\}$, and the sets $\mathcal{Y}_L(\boldsymbol{\phi})$ partition $I\!R^{|L|}$ at fixed $\boldsymbol{\lambda}=(\lambda_1,0)$. If we additionally assume a normal theory linear model, $\boldsymbol{Y}_L|\boldsymbol{\phi}$ has a truncated normal distribution on $\mathcal{Y}_L(\boldsymbol{\phi})$, so means μ_i and variances σ_i^2 also depend on $\left(\boldsymbol{\beta},\sigma^2\right)$. Although the extreme versions of the lasso and elastic net are intractable, the interpretation of Theorem 9 still applies.

6. Numerical Examples

In this Section, both simulated and real data scenarios are presented for the Joint Trained Elastic Net (JT-ENET). The simulation is run with both lucky and unlucky $\boldsymbol{\beta}$ examples. For the ridge regression version of our estimator, the theoretical bound from Proposition 12 implies that a lucky $\boldsymbol{\beta}$ is perpendicular to the unlabeled centroid and a unlucky $\boldsymbol{\beta}$ is parallel to the unlabeled centroid. The result in Theorem 9 presumably extends the generality of this concept. The simulation was designed in part to assess whether the notion of lucky versus unlucky $\boldsymbol{\beta}$ extends to the JT-ENET. The real data sets provide covariate shift applications, so the JT-ENET should have some advantage over supervised learning in terms of a prediction focused objective function on the unlabeled set. It is important to note that only $\boldsymbol{X}_L$, $\boldsymbol{X}_U$, and $\boldsymbol{Y}_L$ were used during training throughout this section.

In all cases, comparisons were made to the supervised elastic net using the R package `glmnet` (Friedman et al., 2010; R Core Team, 2015). This particular implementation is optimized for estimating $\lambda_1+2\lambda_2$ with 10-fold cross validation given $\lambda_1/(\lambda_1+2\lambda_2)$. First, the supervised elastic net was implemented by varying $\lambda_1/(\lambda_1+2\lambda_2)\in[0,1]$ over an equally spaced grid of length 57 to optimize parameters $\boldsymbol{\lambda}$.

Second, the semi-supervised JT-ENET was implemented by estimating its parameters $(\boldsymbol{\lambda}, \boldsymbol{\gamma})$ simultaneously. Calls to the `glmnet` with data augmentations from Proposition 2 were used for all low-level fittings. Parameter $\lambda_1 + 2\lambda_2$ was estimated using 10-fold cross-validation given $\lambda_1/(\lambda_1 + 2\lambda_2)$, γ_1, and γ_2. Parameter $\lambda_1/(\lambda_1 + 2\lambda_2)$ was optimized over the grid $\{0, 0.25, 0.5, 0.75, 1, \hat{a}\}$, where $\hat{a}$ was the optimal supervised setting for this parameter. Fixed grids $\gamma_1 \in \nu^{-1}$ and $\gamma_2 \in \nu$ were used for the other parameters, where $\nu = \{0.1, 0.5, 1, 10, 100, 1000, 10000, \infty\}$ and $\nu^{-1} = \{1/r : r \in \nu\}$. For K-fold cross-validation in the semi-supervised setting, the L cases were partitioned into K folds, $\{L_k\}_{k=1}^K$. Let $\widehat{\boldsymbol{\beta}}_{\boldsymbol{\gamma},\boldsymbol{\lambda}}^{(-k)}$ be the estimate from labeled data $L - L_k$ and unlabeled data $U \cup L_k$, and let the K-fold cross-validated variance be $\widehat{\sigma}_K^2 = \sum_{k=1}^K \left\| \boldsymbol{Y}_{L_k} - \boldsymbol{X}_{L_k} \widehat{\boldsymbol{\beta}}_{\boldsymbol{\gamma},\boldsymbol{\lambda}}^{(-k)} \right\|_2^2 / |L|$. The JT-ENET estimate $\widehat{\boldsymbol{\beta}}_{\hat{\boldsymbol{\gamma}},\hat{\boldsymbol{\lambda}}}$ minimized $\widehat{\sigma}_3^2$ over the grid for $\lambda_1/(\lambda_1 + 2\lambda_2)$, γ_1, and γ_2.

Our objective function was the RMSE on the unlabeled set. The RMSE of $\boldsymbol{X}_U \widehat{\boldsymbol{\beta}}$ from $\boldsymbol{X}_U \boldsymbol{\beta}$ was computed within simulations, but was computed from the withheld responses $\boldsymbol{Y}_U$ in the real data examples. Let ENET and JT-ENET represent this unlabeled set RMSE for the supervised elastic net and our proposed method using the true $\boldsymbol{\beta}$ for the simulations and their empirical versions in real data examples. Percent improvement $\%\text{JT-ENET} = \frac{\text{ENET} - \text{JT-ENET}}{\text{ENET}} \times 100\%$ was used to assess semi-supervised performance. A baseline comparison to the theoretical best parameter settings for the semi-supervised technique was also computed in the simulations, and its percent improvement is denoted by %BEST. Two regression based covariate shift competitors were also applied to the real data examples: adaptive importance-weighted kernel regularized least-squares (AIWKRLS) (Sugiyama et al., 2007) and plain kernel regularized least-squares (PKRLS) (Kananmori et al., 2009). The `caret` package in R (Kuhn, 2008) was also used to fit the SVM with a polynomial kernel on the real data examples.

6.1 Simulations

Same and extrapolated feature data distributions were constructed to study three, high-dimensional scenarios. Each scenario had $|L| = |U| = 100$, $p = 1,000$, true active set $\mathcal{T} = \{1, \ldots, 10\}$, $(\boldsymbol{X}_L)_{ij} \overset{\text{i.i.d.}}{\sim} \text{N}(0, 0.4)$, and $\boldsymbol{Y}_L = \boldsymbol{X}_L \boldsymbol{\beta} + \epsilon$ with $\epsilon \sim \text{N}\left(\vec{0}, \sigma^2 \boldsymbol{I}\right)$. Define indicator vector $\boldsymbol{\mu}(\mathcal{A}) \in I\!R^p$ with entries $\boldsymbol{\mu}_j(\mathcal{A}) = \mathcal{I}_{\{j \in \mathcal{A}\}}$ for some active set $\mathcal{A}$, $\boldsymbol{\beta}^{(\text{unlucky})} = 5\boldsymbol{\mu}(\mathcal{T})/\sqrt{10}$, and $\boldsymbol{\beta}^{(\text{lucky})} = 5\left(\boldsymbol{\mu}(\mathcal{T}_1) - \boldsymbol{\mu}(\mathcal{T}_2)\right)/\sqrt{10}$ with $\mathcal{T}_1 = \{1, \ldots, 5\}$ and $\mathcal{T}_2 = \{6, \ldots, 10\}$. The three scenarios were

1. **Same Distribution:** $(\boldsymbol{X}_U)_{ij} \overset{\text{i.i.d.}}{\sim} \text{N}(0, 0.4)$ and $\boldsymbol{\beta} = \boldsymbol{\beta}^{(\text{lucky})}$
2. **Extrapolation (Lucky $\boldsymbol{\beta}$):** $(\boldsymbol{X}_U)_{ij} \overset{\text{ind}}{\sim} \text{N}\left(10\boldsymbol{\mu}_j(\mathcal{T}), 0.4\right)$ and $\boldsymbol{\beta} = \boldsymbol{\beta}^{(\text{lucky})}$
3. **Extrapolation (Unlucky $\boldsymbol{\beta}$):** $(\boldsymbol{X}_U)_{ij} \overset{\text{ind}}{\sim} \text{N}\left(10\boldsymbol{\mu}_j(\mathcal{T}), 0.4\right)$ and $\boldsymbol{\beta} = \boldsymbol{\beta}^{(\text{unlucky})}$.

If the truth $\boldsymbol{X}_U \boldsymbol{\beta}$ is large, any type of shrinking may be detrimental, so shrinking methods (supervised or semi-supervised) should struggle in the extrapolation scenario with unlucky $\boldsymbol{\beta}$ because $\boldsymbol{\beta} = \boldsymbol{\beta}^{(\text{unlucky})}$ is parallel to the unlabeled data centroid $\boldsymbol{\mu}(\mathcal{T})$. On the other hand, $\boldsymbol{\beta}^{(\text{lucky})} \perp \boldsymbol{\mu}(\mathcal{T})$, so shrinking directions of extrapolation is more desirable. There is an unlucky $\boldsymbol{\beta}$ direction, but a $|\mathcal{T}| - 1$ or 9-dimensional vector space of lucky $\boldsymbol{\beta}$ directions. Setting $\boldsymbol{\beta} = \boldsymbol{\beta}^{(\text{lucky})}$ versus $\boldsymbol{\beta} = \boldsymbol{\beta}^{(\text{unlucky})}$ is not critical in the same distribution scenario.

	Same Distribution			Extrapolation (Lucky $\boldsymbol{\beta}$)			Extrapolation (Unlucky $\boldsymbol{\beta}$)		
σ^2	ENET	%JT-ENET	%BEST	ENET	%JT-ENET	%BEST	ENET	%JT-ENET	%BEST
2.5	0.43	19.27	30.67	0.91	18.08	27.58	15.05	−2.55	2.06
	0.03	3.41	3.63	0.08	3.50	3.65	0.11	0.83	0.65
5.0	0.69	31.88	46.67	1.25	26.10	38.73	15.19	−1.74	2.90
	0.06	4.88	4.43	0.12	4.32	4.43	0.14	0.96	0.86
7.5	0.93	35.36	54.63	1.61	33.49	46.98	15.30	−1.47	4.75
	0.09	5.68	4.58	0.17	4.53	4.50	0.21	1.44	1.25

Table 3: Unlabeled root mean squared error performance is summarized on high-dimensional ($p = 1,000$), simulated data sets: supervised elastic net (ENET), percent improvement over ENET with the joint trained elastic net (%JT-ENET), and the hypothetical maximum of %JT-ENET based on "cheating" with the "answers" $\boldsymbol{X}_U\boldsymbol{\beta}$ while picking the point $(\boldsymbol{\lambda}, \boldsymbol{\gamma})$ in the cross-validation grid (%BEST). Fifty data sets were generated per level combination of scenario (i.e., same, lucky, and unlucky) and model error variance $\sigma^2 = 2.5, 5.0, 7.5$. Cell entries are the sample mean (top) and standard error (bottom).

These probability models were used to conduct simulations studies in the following manner.

Model matrix $\boldsymbol{X}$ was generated once and fixed by scenario, and 50 independent response vectors $\boldsymbol{Y}_L$ were generated from the assumed linear model for each level combination of scenario $= 1, 2, 3$ and $\sigma^2 = 2.5, 5.0, 7.5$. Cross-validation took an average of 3.5 minutes per data set on a 2.6 GHz Intel Core i7 Power Mac. The supervised ENET is best suited for the same distribution prediction task, and its RMSEs are smallest in this scenario. The significant performance advantage due to our semi-supervised adjustment in the same distribution scenario relates to the curse of dimensionality, because extrapolations are likely in a high-dimensional empirical distribution. There was also substantial improvement in the extrapolation with lucky $\boldsymbol{\beta}$ scenario, while both approaches struggled at extrapolation with unlucky $\boldsymbol{\beta}$.

The %BEST values reported in Table 3 correspond to the best possible points $(\boldsymbol{\lambda}, \boldsymbol{\gamma})$ in the cross validation grid and provide at least two points of useful discussion. First, values %BEST increased with σ^2, and this is consistent with what one might expect given the factorization of the bound in Corollary 11. Its left hand side is a nonnegative number that is independent of σ^2, and a semi-supervised improvement is possible when σ^2 exceeds this nonnegative number. The values %BEST in Table 3 supports that a similar concept holds with the bound in Theorem 9 that applies to the JT-ENET. Second, most points in the cross validation grid corresponded to negative percent improvements, and some of these are the largest in magnitude. Thus, while the method of cross validation is not getting the very best point in the grid, its performance is competitive.

6.2 Real Data Examples

The 10 tests listed in Table 4 were constructed using 8 publicly available data sets and a simulated toy extrapolation data set. Each is expected to have a covariate shifted empirical feature data distribution either because the characteristic used to define the labeled set is

Data Set (n, p)	Labeled Set L	Response y	Data Set Source
Toy Cov. Shift $(1200, 1)$	Training Set	$\text{sinc}(x) + \epsilon$	Sugiyama et al. (2007)
Auto-MPG $(398, 8)$	P1: Domestics	Fuel (mpg)	Lichman (2013)
Auto-MPG $(398, 8)$	P2: ≤ 4 Cyl.	Fuel (mpg)	Lichman (2013)
Heart $(462, 8)$	No History	$\sqrt{}$Sys. BP	Hastie et al. (2009)
U.S. News $(1004, 19)$	Private Schools	SAT.ACT	ASA Data Expo '95
Auto-Import $(205, 24)$	Low Risk Cars	Price	Lichman (2013)
Blood Brain $(208, 135)$	Cmpds. 1-52	log(BBB)	Kuhn (2008)
Eye $(120, 200)$	Rats 1-30	$\sqrt{}$Express	Scheetz et al. (2006)
Cookie $(72, 700)$	Training Set	Water	Osborne et al. (1984)
Ethanol $(589, 1037)$	Sols. 1-294	Ethanol	Shen et al. (2013)

Table 4: These ten covariate shift tests are used to establish benchmarks in Table 5.

Data Set	p	$\lvert L\rvert$	$\lvert U\rvert$	ENET	SVM	AIWKRLS	PKRLS	JT-ENET	%JT-ENET
Toy Cov. Shift	1	200	1000	0.527	0.186	**0.103**	0.129	0.169	**67.83**
Auto-MPG (P1)	8	149	249	5.361	5.272	5.974	8.459	**4.341**	**19.02**
Auto-MPG (P2)	8	208	190	8.296	13.478	15.374	39.570	**6.723**	**18.96**
Heart	8	192	270	0.789	0.790	0.795	0.802	**0.788**	**0.13**
U.S. News	19	640	364	1.738	1.724	1.928	1.918	**1.684**	**3.11**
Auto-Import	24	113	92	4995	4223	6292	6376	**4201**	**15.89**
Blood Brain	135	52	156	1.684	6.424	0.797	0.815	**0.649**	**61.46**
Eye	200	30	90	0.019	0.425	0.027	0.027	**0.016**	**15.79**
Cookie	700	40	32	0.388	0.580	1.466	1.309	**0.342**	**11.86**
Ethanol	1037	294	295	1.461	1.422	2.626	2.625	**1.391**	**4.79**

Table 5: Empirical unlabeled root mean squared errors are listed for the ten covariate shift tests defined by Table 4 and a field of five competitors: the supervised elastic net (ENET), a support vector machine (SVM), adaptive importance-weighted kernel regularized least-squares (AIWKRLS), plain kernel regularized least-squares (PKRLS), and joint trained ENET (JT-ENET). The top performer is in bold. The final column is percent improvement of JT-ENET over its supervised ENET alternative with positive values in bold.

associated with other variables in the model matrix, because of the curse of dimensionality, or because the simulated toy data were generated from a model with covariate shift. Since covariate shift is our focus, randomized subsetting of the data (i.e., MCAR) was not performed. When p is larger in the blood brain, eye, cookie, and ethanol applications, the unlabeled set is likely to contain extrapolations. In all cases, the bounds from Section 5 together with the Section 4 geometry of the JT-ENET are at play here behind the scenes. The *U.S. News & World Report* data required preprocessing. SAT scores were transformed to their ACT equivalent, and the new variable with either transformed SAT, ACT, or their average was used instead. Median imputation was used for all other missing values across the board. In the Toy Covariate Shift example, we forced $\boldsymbol{\lambda} = \vec{0}$ for both the ENET and JT-ENET to make comparisons consistent with Sugiyama et al. (2007).

RMSEs for the various approaches and the empirical percent improvement for JT-ENET are reported in Table 5. The JT-ENET appears to have worked in the ideal manner independent of what caused the empirical covariate shift. In their toy covariate shift example, competitors AIWKRLS and PKRLS performed strongly, but their edge went away with increased p. AIWKRLS and PKRLS are principled on estimating empirical density ratios, and this can be a challenging task in practical applications with large p. The SVM and ENET are very close competitors for most of the examples. The results provide further evidence that the JT-ENET is achieving the goal of out-performing the ENET in covariate shift problems.

The JT-ENET fit fairly quickly on a 2.6 GHz Intel Core i7 Power Mac. Thus, if the range of possible improvements is from roughly none to substantial in any given prediction focused application, the associated computational overhead of the JT-ENET appears worthwhile. In addition, it is embarrassingly parallel. Just consider the fixed $6 \times 8 \times 8$ grid search over $(\lambda_1/(\lambda_1 + 2\lambda_2)) \times \gamma_1 \times \gamma_2$ in our implementation. Effective times can essentially be divided by 6 if one sends $1 \times 8 \times 8$ grid searches to each of 6 computers or divided by 48 with grids of $1 \times 1 \times 8$ to 48 computers.

7. Discussion

This work provided a clear and succinct mathematical framework for semi-supervised linear predictions of the unlabeled data. Our joint trained elastic net has two pairs of tuning parameters: supervised $\boldsymbol{\lambda} = (\lambda_1, \lambda_2)$ and semi-supervised $\boldsymbol{\gamma} = (\gamma_1, \gamma_2)$. Adjusting the semi-supervised parameters has an interpretable, geometrical effect on the unlabeled predictions. Furthermore, we provided theoretical bounds for when this interpretable adjustment guarantees a performance improvement under the standard linear model, and this main theme of these theoretical results was validated with simulated data. This practical approach was also competitive with existing approaches throughout a set of challenging, high-dimensional, real data applications, where the unlabeled data contained extrapolations. Extrapolations in the unlabeled set are expected to occur often in practice, due to the curse of dimensionality with large p or practical constraints that result in covariate shift applications, and our method is unique among existing approaches in its direct and effective accounting for these circumstances. Simultaneous estimation of the supervised and semi-supervised tuning parameters was feasible in the high-dimensional examples we tested.

Acknowledgments

The authors thank the AE and three anonymous referees. Their comments and suggestions led to substantial improvements in the presentation of this work. The authors also thank Professor Stephen B. Vardeman. Extensive in-person discussions between the first author and Professor Vardeman led to an understanding of the Joint Training Optimization Problem (6) that ultimately helped both authors articulate its applications. These in-person conversations were made possible through the visiting faculty program within the Statistical Sciences Group at Los Alamos National Laboratory, and the first author is also thankful to be a part of that research program. The work of Mark Vere Culp was supported in part

by the NSF CAREER/DMS-1255045 grant. The opinions and views expressed in this paper are those of the authors and do not reflect the opinions or views at the NSF.

Appendix A. Proofs

Proofs of Propositions and Theorems follow.

A.1 Joint Training Framework

Proposition 2 *If $\gamma_2 > 0$, then rank$(\boldsymbol{X}_U)$ = rank$\left(\boldsymbol{X}_U^{(\gamma_2)}\right)$ and a solution $\widehat{\boldsymbol{\beta}}_{\boldsymbol{\gamma},\boldsymbol{\lambda}}$ to Optimization Problem (9) is a partial solution to Optimization Problem (6).*

Proof Clearly, $\text{rank}\left(\boldsymbol{X}_U\right) = \text{rank}\left(\boldsymbol{X}_U^{(\gamma_2)}\right)$ whenever $\gamma_2 > 0$ by Equation (8). Based on Objective (6), the optimal $\boldsymbol{\alpha}$ at any $\boldsymbol{\beta}$ is $\boldsymbol{\alpha} = \left(\boldsymbol{X}_U^T\boldsymbol{X}_U + \gamma_2\boldsymbol{I}\right)^{-1}\boldsymbol{X}_U^T\boldsymbol{X}_U\boldsymbol{\beta}$ and does not depend on $\gamma_1 > 0$. The derivative with respect to $\boldsymbol{\beta}$ of the objective is proportional to $-\gamma_1\boldsymbol{X}_U^T\boldsymbol{X}_U(\boldsymbol{\alpha}-\boldsymbol{\beta})$ as a function of the unlabeled data, and after plugging-in the optimal $\boldsymbol{\alpha}$ it simplifies to

$$\begin{aligned} -\gamma_1\boldsymbol{X}_U^T\boldsymbol{X}_U(\boldsymbol{\alpha}-\boldsymbol{\beta}) &= -\gamma_1\boldsymbol{X}_U^T\boldsymbol{X}_U\left\{\left(\boldsymbol{X}_U^T\boldsymbol{X}_U+\gamma_2\boldsymbol{I}\right)^{-1}\boldsymbol{X}_U^T\boldsymbol{X}_U - \boldsymbol{I}\right\}\boldsymbol{\beta} \\ &= \gamma_1{\boldsymbol{X}_U^{(\gamma_2)}}^T\boldsymbol{X}_U^{(\gamma_2)}\boldsymbol{\beta}, \end{aligned} \tag{26}$$

where ${\boldsymbol{X}_U^{(\gamma_2)}}^T\boldsymbol{X}_U^{(\gamma_2)} = \gamma_2\boldsymbol{X}_U^T\boldsymbol{X}_U\left(\boldsymbol{X}_U^T\boldsymbol{X}_U+\gamma_2\boldsymbol{I}\right)^{-1}$ used in Equality (26) holds because

$$\begin{aligned} \gamma_2\boldsymbol{X}_U^T\boldsymbol{X}_U &= \gamma_2\boldsymbol{X}_U^T\left(\boldsymbol{X}_U\boldsymbol{X}_U^T+\gamma_2\boldsymbol{I}\right)^{-1}\left(\boldsymbol{X}_U\boldsymbol{X}_U^T+\gamma_2\boldsymbol{I}\right)\boldsymbol{X}_U \\ &= {\boldsymbol{X}_U^{(\gamma_2)}}^T\boldsymbol{X}_U^{(\gamma_2)}\left(\boldsymbol{X}_U^T\boldsymbol{X}_U+\gamma_2\boldsymbol{I}\right). \end{aligned}$$

Thus, the optimal $\widehat{\boldsymbol{\beta}}_{\boldsymbol{\gamma},\boldsymbol{\lambda}}$ from Problem (6) must also solve Problem (9) by Identity (26). ■

A.2 Geometry Results

Proposition 3 *Any eigenbasis of the possibly non-symmetric matrix $\boldsymbol{M}^{(\gamma_2)}$ is real with eigenvalues $\tau_1^{(\gamma_2)} \geq \cdots \geq \tau_p^{(\gamma_2)} \geq 0$. Furthermore, $\tau_i^{(\gamma_2)} = 0$ iff $i >$ rank$(\boldsymbol{X}_U)$.*

Proof Let $\boldsymbol{X}_L^T\boldsymbol{X}_L = \boldsymbol{O}_L\boldsymbol{D}_L\boldsymbol{O}_L^T$ be the eigendecomposition, assume $\text{rank}(\boldsymbol{X}_L) = p$, and define the linear transformation

$$\tilde{\boldsymbol{w}} = \boldsymbol{D}_L^{1/2}\boldsymbol{O}_L^T\boldsymbol{w} \tag{27}$$

that changes the coordinate basis to $\boldsymbol{O}_L$ and then rescales by $\boldsymbol{D}_L^{1/2}$. The symmetric matrix

$$\widetilde{\boldsymbol{M}}^{(\gamma_2)} = \boldsymbol{D}_L^{-1/2}\boldsymbol{O}_L^T{\boldsymbol{X}_U^{(\gamma_2)}}^T\boldsymbol{X}_U^{(\gamma_2)}\boldsymbol{O}_L\boldsymbol{D}_L^{-1/2} \tag{28}$$

has an orthonormal eigenvector decomposition $\left\{\left(\tilde{\boldsymbol{w}}_i^{(\gamma_2)}, \tau_i^{(\gamma_2)}\right)\right\}_{i=1}^p$, so $\boldsymbol{M}^{(\gamma_2)}$ has the real eigendecomposition $\left\{\left(\boldsymbol{w}_i^{(\gamma_2)}, \tau_i^{(\gamma_2)}\right)\right\}_{i=1}^p$ by the reverse of Transformation (27) because

$$\tau_i^{(\gamma_2)}\tilde{\boldsymbol{w}}_i^{(\gamma_2)} = \widetilde{\boldsymbol{M}}^{(\gamma_2)}\tilde{\boldsymbol{w}}_i^{(\gamma_2)} \iff \tau_i^{(\gamma_2)}\boldsymbol{w}_i^{(\gamma_2)} = \boldsymbol{M}^{(\gamma_2)}\boldsymbol{w}_i^{(\gamma_2)}.$$

Furthermore, $\tau_i^{(\gamma_2)} = \tilde{\boldsymbol{w}}_i^{(\gamma_2)^T} \widetilde{\boldsymbol{M}}^{(\gamma_2)} \tilde{\boldsymbol{w}}_i^{(\gamma_2)} = \boldsymbol{w}_i^{(\gamma_2)^T} \boldsymbol{X}_U^{(\gamma_2)^T} \boldsymbol{X}_U^{(\gamma_2)} \boldsymbol{w}_i^{(\gamma_2)} = 0$ iff $\tau_i^{(\gamma_2)} = 0$. ■

Proposition 4 *The path $\widehat{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}$ as a function of $\gamma_1 \geq 0$ is bounded within a p-dimensional parallelotope with corners at each binary linear combination of $\left\{\hat{c}_1^{(\gamma_2)} \boldsymbol{w}_1^{(\gamma_2)}, \ldots, \hat{c}_p^{(\gamma_2)} \boldsymbol{w}_p^{(\gamma_2)}\right\}$. Furthermore, the terminal point as $\gamma_1 \to \infty$ is the corner $\sum_{i=1}^p \mathcal{I}_{\{i > rank(\boldsymbol{X}_U)\}} \hat{c}_i^{(\gamma_2)} \boldsymbol{w}_i^{(\gamma_2)}$ with indicator $\mathcal{I}_{\{\cdot\}}$.*

Proof Decomposing $\widehat{\boldsymbol{\beta}}^{(\text{OLS})}$ in Equation (15) onto the real eigenbasis $\left\{\boldsymbol{w}_i^{(\gamma_2)}\right\}_{i=1}^p$ from Proposition 2 and then applying Equation (14) to establish Equation (16) are the main steps. Path $\widehat{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}$ goes to the terminal point as $\gamma_1 \to \infty$ because the probability weights $1/\left(1 + \gamma_1 \tau_i^{(\gamma_2)}\right)$ in Equation (16) have limits of 0 or 1 when $\tau_i^{(\gamma_2)} > 0$ or $\tau_i^{(\gamma_2)} = 0$. Next, consider the set of all vectors within the p-dimensional parallelotope defined by each binary linear combination of $\left\{\hat{c}_i^{(\gamma_2)} \boldsymbol{w}_i^{(\gamma_2)}\right\}_{i=1}^p$ and those for the p-dimensional rectangle defined by each binary linear combination of $\left\{\hat{c}_i^{(\gamma_2)} \tilde{\boldsymbol{w}}_i^{(\gamma_2)}\right\}_{i=1}^p$, where $\left\{\tilde{\boldsymbol{w}}_i^{(\gamma_2)}\right\}_{i=1}^p$ are orthonormal eigenvectors of Matrix (28). Transformation (27) is a bijection from the parallelotope to the rectangle. This bijective mapping replaces the $\boldsymbol{w}_i^{(\gamma_2)}$ on the right of Equation (16) with $\tilde{\boldsymbol{w}}_i^{(\gamma_2)}$, and so $\widehat{\boldsymbol{\beta}}_{\boldsymbol{\gamma}} \mapsto \boldsymbol{D}_L^{1/2} \boldsymbol{O}_L^T \widehat{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}$ is clearly within the rectangle. ■

Proposition 5 *The* $\text{span}\left(\boldsymbol{X}^{(\gamma_2)^T} \boldsymbol{X}^{(\gamma_2)} \boldsymbol{w}_i^{(\gamma_2)}\right) = \bigcap_{j \in \{1,\ldots,p\} - \{i\}} \boldsymbol{w}_j^{(\gamma_2)^\perp}$ $\forall i \in \{1, \ldots, p\}$. *Henceforth, the line* $\text{span}\left(\boldsymbol{X}^{(\gamma_2)^T} \boldsymbol{X}^{(\gamma_2)} \boldsymbol{w}_i^{(\gamma_2)}\right)$ *is called the i^{th} extrapolation direction $\forall i \in \{1, \ldots, p\}$.*

Proof If $\left\{\tilde{\boldsymbol{w}}_i^{(\gamma_2)}\right\}_{i=1}^p$ are orthonormal eigenvectors of the Symmetric Matrix (28),

$$\boldsymbol{w}_i^{(\gamma_2)^T} \boldsymbol{X}_L^T \boldsymbol{X}_L \boldsymbol{w}_j^{(\gamma_2)} = \mathcal{I}_{\{i=j\}} \tag{29}$$

$$\boldsymbol{w}_i^{(\gamma_2)^T} \boldsymbol{X}_U^{(\gamma_2)^T} \boldsymbol{X}_U^{(\gamma_2)} \boldsymbol{w}_j^{(\gamma_2)} = \mathcal{I}_{\{i=j\}} \tau_i^{(\gamma_2)} \tag{30}$$

by Transformation (27). Let $\boldsymbol{\nu} \in \text{span}\left(\boldsymbol{X}^{(\gamma_2)^T} \boldsymbol{X}^{(\gamma_2)} \boldsymbol{w}_i^{(\gamma_2)}\right)$. Summing Equations (29) and (30) implies that $\boldsymbol{\nu}^T \boldsymbol{w}_j^{(\gamma_2)} = 0$ and hence $\boldsymbol{\nu} \in \boldsymbol{w}_j^{(\gamma_2)^\perp}$ for each $j \neq i$. Now, let $\boldsymbol{\nu} \in \bigcap_{j \neq i} \boldsymbol{w}_j^{(\gamma_2)^\perp} \subseteq \mathbb{R}^p$, so $\boldsymbol{\nu}^T \boldsymbol{w}_j^{(\gamma_2)} = 0$ for each $j \neq i$. There exists a unique sequence $\{a_k\}_{k=1}^p$ such that $\boldsymbol{\nu} = \sum_{k=1}^p a_k \boldsymbol{X}^{(\gamma_2)^T} \boldsymbol{X}^{(\gamma_2)} \boldsymbol{w}_k^{(\gamma_2)}$ by the assumption $\text{rank}(\boldsymbol{X}_L) = p$, so $\boldsymbol{\nu}^T \boldsymbol{w}_j^{(\gamma_2)} = a_j \left(1 + \tau_j^{(\gamma_2)}\right)$ by Equations (29) and (30). Thus, $a_j = 0$ for each $j \neq i$ and $\boldsymbol{\nu} \in \text{span}\left(\boldsymbol{X}^{(\gamma_2)^T} \boldsymbol{X}^{(\gamma_2)} \boldsymbol{w}_i^{(\gamma_2)}\right)$. ■

Proposition 6 *If $\gamma_2 > 0$, vectors $\left\{\boldsymbol{\ell}_i^{(\gamma_2)}\right\}_{i=p}^1$ and $\left\{\boldsymbol{u}_i^{(\gamma_2)}\right\}_{i=1}^{rank(\boldsymbol{X}_U)}$ are orthonormal bases for the column spaces of $\boldsymbol{X}_L$ and $\boldsymbol{X}_U^{(\gamma_2)}$, and $\boldsymbol{u}_i^{(\gamma_2)} = \vec{0}$ if $i > rank(\boldsymbol{X}_U)$.*

Proof The orthonormality holds by Definitions (17) and Identities (29) and (30). Note $\boldsymbol{u}_i^{(\gamma_2)} = \vec{0}$ if $i > \text{rank}(\boldsymbol{X}_U)$ by Identity (30). The column space result follows from Equation (19) and the joint trained least squares assumption of $\text{rank}(\boldsymbol{X}_L) = p$. ■

Proposition 7 *For each $i \in \{1,\ldots,p\}$,*

$$\begin{aligned}\boldsymbol{X}_L^T\boldsymbol{\ell}_i^{(\gamma_2)} &= \frac{1}{1+\tau_i^{(\gamma_2)}}\boldsymbol{X}^{(\gamma_2)^T}\boldsymbol{X}^{(\gamma_2)}\boldsymbol{w}_i^{(\gamma_2)}\\ \boldsymbol{X}_U^{(\gamma_2)^T}\boldsymbol{u}_i^{(\gamma_2)} &= \frac{\tau_i^{(\gamma_2)}}{(1+\tau_i^{(\gamma_2)})\sqrt{\kappa_i^{(\gamma_2)}}}\boldsymbol{X}^{(\gamma_2)^T}\boldsymbol{X}^{(\gamma_2)}\boldsymbol{w}_i^{(\gamma_2)},\end{aligned}$$

so $\boldsymbol{X}^{(\gamma_2)^T}\boldsymbol{X}^{(\gamma_2)}\boldsymbol{w}_i^{(\gamma_2)}$, $\boldsymbol{X}_L^T\boldsymbol{\ell}_i^{(\gamma_2)}$, and $\boldsymbol{X}_U^{(\gamma_2)^T}\boldsymbol{u}_i^{(\gamma_2)}$ are parallel vectors in $\mathbb{R}^p$.

Proof By Definitions (8), (14), and (17),

$$\begin{aligned}\tau_i^{(\gamma_2)}\boldsymbol{w}_i^{(\gamma_2)} &= \boldsymbol{M}^{(\gamma_2)}\boldsymbol{w}_i^{(\gamma_2)}\\ \tau_i^{(\gamma_2)}\boldsymbol{X}_L^T\boldsymbol{X}_L\boldsymbol{w}_i^{(\gamma_2)} &= \boldsymbol{X}_U^{(\gamma_2)^T}\boldsymbol{X}_U^{(\gamma_2)}\boldsymbol{w}_i^{(\gamma_2)} &(31)\\ \tau_i^{(\gamma_2)}\boldsymbol{X}_L^T\boldsymbol{\ell}_i^{(\gamma_2)} &= \sqrt{\kappa_i^{(\gamma_2)}}\boldsymbol{X}_U^{(\gamma_2)^T}\boldsymbol{u}_i^{(\gamma_2)} &(32)\\ \left(1+\tau_i^{(\gamma_2)}\right)\boldsymbol{X}_L^T\boldsymbol{\ell}_i^{(\gamma_2)} &= \boldsymbol{X}^{(\gamma_2)^T}\boldsymbol{X}^{(\gamma_2)}\boldsymbol{w}_i^{(\gamma_2)}. &(33)\end{aligned}$$

Hence, Vectors (31)-(33) are parallel, and the stated identities follow from Equations (32) and (33). ■

A.3 Performance Bounds

Theorem 9 *Let Assumptions 1-3 hold. Also, let $q \geq 1$, $\tau_1^{(\boldsymbol{\phi})} > 0$, and $p_i\left(\boldsymbol{\tau}^{(\boldsymbol{\phi})}\right) = \frac{\tau_i^{(\boldsymbol{\phi})^2}\sigma_i^2}{\sum_{j=1}^q \tau_j^{(\boldsymbol{\phi})^2}\sigma_j^2}$. If $\sum_{i=1}^q p_i\left(\boldsymbol{\tau}^{(\boldsymbol{\phi})}\right)\left(\frac{\mu_i\left(c_i^{(\boldsymbol{\phi})}+\boldsymbol{u}_i^{(\boldsymbol{\phi})^T}\boldsymbol{X}_{U\bar{\mathcal{A}}}\boldsymbol{\beta}_{\bar{\mathcal{A}}}/\sqrt{\kappa_i^{(\gamma_2)}}\right)-\mu_i^2}{\sigma_i^2}\right) < 1$, then*

$$\mathbb{E}\left[\left\|\boldsymbol{X}_U\left(\widehat{\boldsymbol{\beta}}_{\gamma_1}^{(\boldsymbol{\phi})}-\boldsymbol{\beta}\right)\right\|_2^2\middle|\boldsymbol{\phi}\right] < \mathbb{E}\left[\left\|\boldsymbol{X}_U\left(\widehat{\boldsymbol{\beta}}_{\boldsymbol{\lambda}}^{(SUP)}-\boldsymbol{\beta}\right)\right\|_2^2\middle|\boldsymbol{\phi}\right].$$

Proof Let $\gamma_1 \in [0,\delta)$ for $\delta > 0$ from Assumption 2, and define $\boldsymbol{u}_i^{(\boldsymbol{\phi})} = \boldsymbol{X}_U\boldsymbol{w}_i^{(\boldsymbol{\phi})}/\sqrt{\kappa_i^{(\boldsymbol{\phi})}}$, where $\kappa_i^{(\boldsymbol{\phi})} = \tau_i^{(\boldsymbol{\phi})} + \mathcal{I}_{\{i>\mathrm{rank}(\boldsymbol{X}_U)\}} > 0$ and hence $\kappa_i^{(\boldsymbol{\phi})}\tau_i^{(\boldsymbol{\phi})} = \tau_i^{(\boldsymbol{\phi})^2}$. Vectors $\left\{\boldsymbol{u}_i^{(\boldsymbol{\phi})}\right\}_{i=1}^q$ are an orthonormal basis for the column space of $\boldsymbol{X}_U$ by Proposition 6, and

$$\boldsymbol{X}_{U\mathcal{A}}\left(\left(\widehat{\boldsymbol{\beta}}_{\gamma_1}^{(\boldsymbol{\phi})}\right)_{\mathcal{A}}-\boldsymbol{\beta}_{\mathcal{A}}\right) = \sum_{i=1}^q\left(\frac{\hat{c}_i^{(\boldsymbol{\phi})}}{1+\gamma_1\tau_i^{(\boldsymbol{\phi})}}-c_i^{(\boldsymbol{\phi})}\right)\boldsymbol{u}_i^{(\boldsymbol{\phi})}\sqrt{\kappa_i^{(\boldsymbol{\phi})}} \quad (34)$$

by Equations (24) and (25). Next, define loss function

$$Q = \left\|\boldsymbol{X}_U\left(\widehat{\boldsymbol{\beta}}_{\gamma_1}^{(\boldsymbol{\phi})}-\boldsymbol{\beta}\right)\right\|_2^2 = Q_1+Q_2+Q_3, \quad (35)$$

where $Q_1 = \left\|\boldsymbol{X}_{U\mathcal{A}}\left(\left(\widehat{\boldsymbol{\beta}}_{\gamma_1}^{(\boldsymbol{\phi})}\right)_{\mathcal{A}}-\boldsymbol{\beta}_{\mathcal{A}}\right)\right\|_2^2$, $Q_2 = -2\left(\left(\widehat{\boldsymbol{\beta}}_{\gamma_1}^{(\boldsymbol{\phi})}\right)_{\mathcal{A}}-\boldsymbol{\beta}_{\mathcal{A}}\right)^T\boldsymbol{X}_{U\mathcal{A}}^T\boldsymbol{r}$, $Q_3 = \|\boldsymbol{r}\|_2^2$, and $\boldsymbol{r} = \boldsymbol{X}_{U\bar{\mathcal{A}}}\boldsymbol{\beta}_{\bar{\mathcal{A}}}$. If $\gamma_1 = 0$, the supervised estimator $\widehat{\boldsymbol{\beta}}_{\boldsymbol{\lambda}}^{(\mathrm{SUP})}$ follows, so an improvement is guaranteed if the gradient of $\mathbb{E}\left[Q|\boldsymbol{\phi}\right]$ with respect to γ_1 evaluated at $\gamma_1 = 0$ is negative.

By Equation (34) and Assumption 3,

$$\begin{aligned}\mathbb{E}\left[Q_1|\boldsymbol{\phi}\right] &= \sum_{i=1}^{q}\mathbb{E}\left[\left(\frac{\hat{c}_i^{(\boldsymbol{\phi})}}{1+\gamma_1\tau_i^{(\boldsymbol{\phi})}}-c_i^{(\boldsymbol{\phi})}\right)^2\middle|\boldsymbol{\phi}\right]\kappa_i^{(\boldsymbol{\phi})} \\ &= \sum_{i=1}^{q}\left(\left(\frac{1}{1+\gamma_1\tau_i^{(\boldsymbol{\phi})}}\right)^2\left(\sigma_i^2+\mu_i^2\right)-2\frac{\mu_i}{1+\gamma_1\tau_i^{(\boldsymbol{\phi})}}c_i^{(\boldsymbol{\phi})}+{c_i^{(\boldsymbol{\phi})}}^2\right)\kappa_i^{(\boldsymbol{\phi})}, \qquad (36)\end{aligned}$$

and the gradient of Equation (36) is

$$\frac{\partial\mathbb{E}\left[Q_1|\phi\right]}{\partial\gamma_1} = -2\sum_{i=1}^{q}\frac{\tau_i^{(\boldsymbol{\phi})}\kappa_i^{(\boldsymbol{\phi})}}{\left(1+\gamma_1\tau_i^{(\boldsymbol{\phi})}\right)^3}\left(\sigma_i^2+\mu_i^2-c_i^{(\boldsymbol{\phi})}\mu_i-\gamma_1\mu_i c_i^{(\boldsymbol{\phi})}\tau_i^{(\boldsymbol{\phi})}\right). \qquad (37)$$

Similarly for the second term Q_2 on the right of Equation (35),

$$\begin{aligned}\mathbb{E}\left[Q_2|\boldsymbol{\phi}\right] &= -2\sum_{i=1}^{q}\left(\frac{\mu_i}{1+\gamma_1\tau_i^{(\boldsymbol{\phi})}}-c_i^{(\boldsymbol{\phi})}\right)\sqrt{\kappa_i^{(\boldsymbol{\phi})}}{\boldsymbol{u}_i^{(\boldsymbol{\phi})}}^T\boldsymbol{r} \\ \frac{\partial\mathbb{E}\left[Q_2|\phi\right]}{\partial\gamma_1} &= 2\sum_{i=1}^{q}\frac{\tau_i^{(\boldsymbol{\phi})}\sqrt{\kappa_i^{(\boldsymbol{\phi})}}\mu_i}{\left(1+\gamma_1\tau_i^{(\boldsymbol{\phi})}\right)^2}{\boldsymbol{u}_i^{(\boldsymbol{\phi})}}^T\boldsymbol{r}. \qquad (38)\end{aligned}$$

The third term Q_3 on the right of Equation (35) is constant with respect to γ_1 and thus ignored, and the sum of Scores (37) and (38) with $\gamma_1 = 0$ is negative whenever

$$\begin{aligned}-2\sum_{i=1}^{q}\tau_i^{(\boldsymbol{\phi})}\kappa_i^{(\boldsymbol{\phi})}\left(\sigma_i^2+\mu_i^2-c_i^{(\boldsymbol{\phi})}\mu_i-\mu_i{\boldsymbol{u}_i^{(\boldsymbol{\phi})}}^T\boldsymbol{r}/\sqrt{\kappa_i^{(\boldsymbol{\phi})}}\right) &< 0 \\ \sum_{i=1}^{q}{\tau_i^{(\boldsymbol{\phi})}}^2\left(\mu_i\left(c_i^{(\boldsymbol{\phi})}+{\boldsymbol{u}_i^{(\boldsymbol{\phi})}}^T\boldsymbol{r}/\sqrt{\kappa_i^{(\boldsymbol{\phi})}}\right)-\mu_i^2\right) &< \sum_{i=1}^{q}{\tau_i^{(\boldsymbol{\phi})}}^2\sigma_i^2 \\ \sum_{i=1}^{q}p_i\left(\boldsymbol{\tau}^{(\boldsymbol{\phi})}\right)\left(\mu_i\left(c_i^{(\boldsymbol{\phi})}+{\boldsymbol{u}_i^{(\boldsymbol{\phi})}}^T\boldsymbol{r}/\sqrt{\kappa_i^{(\boldsymbol{\phi})}}\right)-\mu_i^2\right)/\sigma_i^2 &< 1.\end{aligned}$$

■

Proposition 12 *If $\tau_1^{(\boldsymbol{\phi})} > 0$ and $\boldsymbol{\beta}_i \in \bigcap_{j\in\{1,\ldots,p\}-\{i\}}{\boldsymbol{w}_j^{(\boldsymbol{\phi})}}^{\perp}$ is unit length, then the joint trained ridge performance bound from Corollary 11 satisfies $\sigma_{LB}^2\left(\boldsymbol{\beta}_i\right) \geq \lambda_2 p_i\left(\boldsymbol{\tau}^{(\boldsymbol{\phi})}\right)$ for $i \in \{1,\ldots,p\}$ and $\sigma_{LB}^2\left(\boldsymbol{\beta}_i\right) \geq \sigma_{LB}^2\left(\boldsymbol{w}_j^{(\boldsymbol{\phi})}/\left\|\boldsymbol{w}_j^{(\boldsymbol{\phi})}\right\|_2\right)$ if $j \geq i$.*

Proof The desired vectors are $\boldsymbol{\beta}_j = {\boldsymbol{X}_L^{(\lambda_2)}}^T\boldsymbol{\ell}_i^{(\boldsymbol{\phi})}/\left\|{\boldsymbol{X}_L^{(\lambda_2)}}^T\boldsymbol{\ell}_i^{(\boldsymbol{\phi})}\right\|_2$ by Proposition 5, so

$${\boldsymbol{w}_i^{(\boldsymbol{\phi})}}^T\boldsymbol{\beta}_j = \mathcal{I}_{\{i=j\}}/\left\|{\boldsymbol{X}_L^{(\lambda_2)}}^T\boldsymbol{\ell}_i^{(\boldsymbol{\phi})}\right\|_2 \qquad (39)$$

by Equation (29). Constraints (39) imply that only term $i = j$ of $\sigma_{\text{LB}}^2(\boldsymbol{\beta}_j)$ can be nonzero. For any $\boldsymbol{\beta} \in \mathbb{R}^p$, $\boldsymbol{\beta} = \sum_{i=1}^{p} c_i^{(\boldsymbol{\phi})}\boldsymbol{w}_i^{(\boldsymbol{\phi})}$ with $c_i^{(\boldsymbol{\phi})} = {\boldsymbol{w}_i^{(\boldsymbol{\phi})}}^T{\boldsymbol{X}_L^{(\lambda_2)}}^T\boldsymbol{X}_L^{(\lambda_2)}\boldsymbol{\beta}$ by Equation (29), so

$c_j^{(\phi)} = \left\| \boldsymbol{X}_L^{(\lambda_2)^T} \boldsymbol{\ell}_j^{(\phi)} \right\|_2$ if $\boldsymbol{\beta} = \boldsymbol{\beta}_j$. These facts can help simplify the bound to

$$\sigma^2_{\mathrm{LB}}(\boldsymbol{\beta}_j) \;=\; \lambda_2 p_j \left(\boldsymbol{\tau}^{(\phi)}\right) \left(1 + \lambda_2 \frac{\left(\boldsymbol{w}_j^{(\phi)^T} \boldsymbol{w}_j^{(\phi)} - 1/ \left\| \boldsymbol{X}_L^{(\lambda_2)^T} \boldsymbol{\ell}_j^{(\phi)} \right\|_2^2 \right)}{\boldsymbol{w}_j^{(\phi)^T} \boldsymbol{X}_L^T \boldsymbol{X}_L \boldsymbol{w}_j^{(\phi)}} \right)_+ . \tag{40}$$

Next, define $\boldsymbol{G} = \left[\boldsymbol{w}_i^{(\phi)^T} \boldsymbol{w}_j^{(\phi)} \right]_{i,j=1}^p$ as the Gram matrix of vectors $\boldsymbol{w}_i^{(\phi)}$. Let $\boldsymbol{G}^{(-j)}$ be the $(p-1) \times (p-1)$ sub matrix of $\boldsymbol{G}$ obtained by deleting the j^{th} row and column, and let $\boldsymbol{G}_j$ be the $1 \times (p-1)$ vector obtained by deleting the j^{th} entry from the j^{th} row of $\boldsymbol{G}$. Matrix $\boldsymbol{G}^{(-j)}$ is positive definite by Proposition 3, and it can be shown that $\left(\boldsymbol{w}_j^{(\phi)^T} \boldsymbol{w}_j^{(\phi)} - 1/ \left\| \boldsymbol{X}_L^{(\lambda_2)^T} \boldsymbol{\ell}_j^{(\phi)} \right\|_2^2 \right) = \boldsymbol{G}_j^T \left(\boldsymbol{G}^{(-j)} \right)^{-1} \boldsymbol{G}_j \geq 0$ by Constraints (39). Therefore, Bound (40) further reduces to

$$\sigma^2_{\mathrm{LB}}(\boldsymbol{\beta}_j) \;=\; \lambda_2 p_j \left(\boldsymbol{\tau}^{(\phi)}\right) \left(1 + \lambda_2 \frac{\boldsymbol{G}_j^T \left(\boldsymbol{G}^{(-j)} \right)^{-1} \boldsymbol{G}_j}{\boldsymbol{w}_j^{(\phi)^T} \boldsymbol{X}_L^T \boldsymbol{X}_L \boldsymbol{w}_j^{(\phi)}} \right) \geq \lambda_2 p_j \left(\boldsymbol{\tau}^{(\phi)}\right). \tag{41}$$

For the second part, define $\nu_{ij} = \frac{\left(\boldsymbol{w}_j^{(\phi)^T} \boldsymbol{w}_i^{(\phi)} \right)^2}{\left\| \boldsymbol{w}_i^{(\phi)} \right\|_2^2 \boldsymbol{w}_i^{(\phi)^T} \boldsymbol{X}_L^T \boldsymbol{X}_L \boldsymbol{w}_i^{(\phi)}} \geq 0$, so

$$\sigma^2_{\mathrm{LB}} \left(\boldsymbol{w}_j^{(\phi)} / \left\| \boldsymbol{w}_j^{(\phi)} \right\|_2 \right) \;=\; \left(\lambda_2 p_j \left(\boldsymbol{\tau}^{(\phi)}\right) - \lambda_2^2 \sum_{i \neq j} p_j \left(\boldsymbol{\tau}^{(\phi)}\right) \nu_{ij} \right)_+ . \tag{42}$$

The result is trivial if Bound (42) is zero, and the difference of Bounds (41) and (42)

$$\sigma^2_{\mathrm{LB}}(\boldsymbol{\beta}_i) - \sigma^2_{\mathrm{LB}} \left(\boldsymbol{w}_j^{(\phi)} / \left\| \boldsymbol{w}_j^{(\phi)} \right\|_2 \right) \geq \lambda_2 \left(p_i \left(\boldsymbol{\tau}^{(\phi)}\right) - p_j \left(\boldsymbol{\tau}^{(\phi)}\right) \right) + \lambda_2^2 \sum_{i \neq j} p_j \left(\boldsymbol{\tau}^{(\phi)}\right) \nu_{ij}$$

is no less than the sum of two non-negative terms if Bound (42) is positive and $j \geq i$. ■

References

A Aswani, P Bickel, and C Tomlin. Regression on manifolds: estimation of the exterior derivative. *The Annals of Statistics*, 39(1):48–81, 2010.

M Azizyan, A Singh, and L Wasserman. Density-sensitive semisupervised inference. *The Annals of Statistics*, 41(2):751–771, 2013.

M Belkin, P Niyogi, and V Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7:2399–2434, 2006.

G Casella. Minimax ridge regression estimation. *The Annals of Statistics*, 8:937–1178, 1980.

O Chapelle, M Chi, and A Zien. A continuation method for semi-supervised SVMs. In *International Conference on Machine Learning*, 2006a.

O Chapelle, B Schölkopf, and A Zien, editors. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006b. URL `http://www.kyb.tuebingen.mpg.de/ssl-book`.

M Culp. On the semi-supervised joint trained elastic net. *Journal of Computational Graphics and Statistics*, 22(2):300–318, 2013.

J Friedman, T Hastie, and R Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010.

A Gretton, A Smola, J Huang, M Schmittfull, K Borgwardt, and B Schölkopf. Covariate shift by kernel mean matching. In J Quiñonero-Candela, M Sugiyama, A Schwaighofer, and N Lawrence, editors, *Dataset Shift in Machine Learning*, pages 1–38. The MIT Press, 2009.

T Hastie, R Tibshirani, and J Friedman. *The Elements of Statistical Learning (Data Mining, Inference, and Prediction)*. Springer Verlag, 2009.

M Hein, J Audibert, and U von Luxburg. From graphs to manifolds–weak and strong pointwise consistency of graph Laplacians. In *Conference on Learning Theory*, pages 470–485, 2005.

T Kananmori, S Hido, and M Sugiyama. A least-squares approach to direct importance estimation. *Journal of Machine Learning Research*, 10:1391–1445, 2009.

M Kuhn. Building predictive models in R using the `caret` package. *Journal of Statistical Software*, 28(5):1–26, 2008.

J Lafferty and L Wasserman. Statistical analysis of semi-supervised regression. In *Advances in NIPS*, pages 801–808. MIT Press, 2007.

M Lichman. UCI machine learning repository, 2013. URL `http://archive.ics.uci.edu/ml`.

S Mente and F Lombardo. A recursive-partitioning model for blood-brain barrier permeation. *Journal of Computer-Aided Molecular Design*, 19:465–481, 2005.

J Moreno-Torres, T Raeder, R Alaiz-Rodrguez, N Chawla, and F Herrera. A unifying view on dataset shift in classification. *Pattern Recognition*, 45(1):521–530, 2008.

BG Osborne, T Fearn, AR Miller, and S Douglas. Application of near infrared reflectance spectroscopy to the compositional analysis of biscuits and biscuit doughs. *Journal of the Science of Food and Agriculture*, 35:99–105, 1984.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, 2015. URL `http://www.R-project.org/`.

T Scheetz, K Kim, R Swiderski, A Philp, T Braun, K Knudtson, A Dorrance, G DiBona, J Huang, T Casavant, V Sheffield, and E Stone. Regulation of gene expression in the mammalian eye and its relevance to eye disease. *Proceedings of the National Academy of Sciences*, 103(39):14429–14434, 2006.

X Shen, M Alam, F Fikse, and L Rönnegård. A novel generalized ridge regression method for quantitative genetics. *Genetics*, 193(4):1255–1268, 2013. URL `http://www.genetics.org/content/193/4/1255.full`.

M Sugiyama, M Krauledat, and K Müller. Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, 8:985–1005, 2007.

J Wang, X Shen, and W Pan. On efficient large margin semisupervised learning: Method and theory. *Journal of Machine Learning Research*, 10:719–742, 2009.

J Wang, T Jebara, and S Chang. Semi-supervised learning using greedy max-cut. *Journal of Machine Learning Research*, 14:771–800, 2013.

K Yamazaki, M Kawanabe, S Watanabe, M Sugiyama, and K Müller. Asymptotic Bayesian generalization error when training and test distributions are different. In *Proceedings of the 24th International Conference on Machine Learning*, pages 1079–1086, 2007.

X Zhu and A Goldberg. *Introduction to Semi-Supervised Learning*. Morgan and Claypool Publishers, 2009.

H Zou and T Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society Series B*, 67(2):301–320, 2005.

Journal of Machine Learning Research 16 (2015) 3219-3267 Submitted 2/14; Revised 4/15; Published 12/15

Ultra-Scalable and Efficient Methods for Hybrid Observational and Experimental Local Causal Pathway Discovery

Alexander Statnikov ALEXANDER.STATNIKOV@MED.NYU.EDU
Sisi Ma SISI.MA@NYUMC.ORG
Mikael Henaff MBH305@NYU.EDU
Nikita Lytkin NIKITA.LYTKIN@GMAIL.COM
Efstratios Efstathiadis STRATOS@NYU.ORG
Eric R. Peskin ERIC.PESKIN@NYUMC.ORG
Center for Health Informatics and Bioinformatics
New York University School of Medicine
New York, NY 10016, USA
Constantin F. Aliferis CALIFERI@UMN.EDU
Institute for Health Informatics
University of Minnesota
Minneapolis, MN 55455, USA

Editor: Peter Sprites

Abstract

Discovery of causal relations from data is a fundamental objective of several scientific disciplines. Most causal discovery algorithms that use observational data can infer causality only up to a statistical equivalency class, thus leaving many causal relations undetermined. In general, complete identification of causal relations requires experimentation to augment discoveries from observational data. This has led to the recent development of several methods for active learning of causal networks that utilize both observational and experimental data in order to discover causal networks. In this work, we focus on the problem of discovering local causal pathways that contain only direct causes and direct effects of the target variable of interest and propose new discovery methods that aim to minimize the number of required experiments, relax common sufficient discovery assumptions in order to increase discovery accuracy, and scale to high-dimensional data with thousands of variables. We conduct a comprehensive evaluation of new and existing methods with data of dimensionality up to 1,000,000 variables. We use both artificially simulated networks and *in-silico* gene transcriptional networks that model the characteristics of real gene expression data.

Keywords: causality, large-scale experimental design, local causal pathway discovery, observational data, experimental data, randomized experiments

1. Introduction

Discovery of causal relations from data is a fundamental objective of several scientific disciplines including computer science, statistics, and applied mathematics (Pearl, 2009; Spirtes et al., 2000; Neapolitan, 2003; Pearl, 1997). Obtaining data from randomized controlled experiments, while being essential for the discovery of causality, is very expensive and is often infeasible or unethical. On the other hand, observational data that is collected without experimental interference of the values of variables is highly abundant and can often be collected cheaply. Over the last 20 years, many sound algorithms have been proposed that can use observational data to infer causal relations (Pearl, 2009; Spirtes et al., 2000; Glymour and Cooper, 1999) and several empirical studies have verified their applicability and scalability to high-dimensional data (Aliferis et al., 2010a,b). However, observational data is, in general, insufficient to completely unravel all causal relations among measured variables, because many causal relations cannot be statistically distinguished with observational data alone (e.g., multiple graphs in the Markov equivalence class). Therefore, it is essential to refine discoveries from observational data with limited and targeted experimental data (Spirtes et al., 2000). This has led to the recent development of several methods for *active learning of causal networks* that utilize observational and experimental data in order to discover causal networks (Tong and Koller, 2001; Murphy, 2001; He and Geng, 2008; Meganck et al., 2006; Hyttinen et al., 2010; Eberhardt et al., 2010; Hyttinen et al., 2012; Pe'er et al., 2001; Sachs et al., 2005).

The present work is concerned with the problem of discovery of local causal pathways that only contain direct causes and direct effects of the target variable of interest, rather than learning the structure of the entire causal network that represents all causal relations among all measured variables. Knowledge of direct causes and effects is crucial for understanding the mechanisms of causality, and knowledge of direct causes particularly facilitates the design of effective interventions. Existing methods for discovery of local causal pathways fully rely on observational data and can discover causality up to a Markov equivalence class, leaving many causal relations undetermined (Spirtes et al., 2000; Aliferis et al., 2010a). Thus, experimental/manipulated data is needed to complement the discovery from observational data. For experimental/manipulated data, we consider here only data from fully randomized experiments (also known as *surgical* or *edge-breaking*). In the present study, all decisions about edge orientation are based on experimental data exclusively. It is noteworthy that the problem of local causal pathway discovery from observational and limited experimental data has not been addressed in the literature previously.

While developing new methods for local causal pathway discovery from observational and experimental data, we set four objectives. First, to *minimize the number of experiments* needed to refine discoveries from observational data. Second, to *relax sufficient assumptions of existing discovery methods* in order to take into account multiplicity of local causal pathways consistent with the data (Statnikov et al., 2013; Statnikov and Aliferis, 2010). The latter has potential to reduce the number of false negative and false positives predictions and improve overall discovery accuracy. Third, to *scale to very high-dimensional data* with many thousands of variables. Finally fourth, to *achieve sufficiently good structure discovery performance*.

As a result of this work, we introduce new ultra-scalable and experimentally efficient

local causal pathway discovery methods and conduct a comprehensive evaluation of new and existing techniques with high-dimensional data with up to 1,000,000 variables. We use both artificially simulated networks and *in-silico* gene transcriptional networks that model the characteristics of real gene expression data. In the latter networks, we focus on discovery of local causal transcriptional pathways of genes. Learning transcriptional pathways is one of the key problems in biomedicine and is a major component of the efforts to develop new diagnostics, vaccines and therapies that will diagnose, prevent and treat deadly human diseases.

The remainder of the paper is organized as follows. Section 2 provides general theory and background. Section 3 provides an overview and discussion of prior methods for active learning of causal networks and how these methods were applied in our study. Section 4 introduces new methods for local causal pathway discovery from observational and experimental data. Section 5 describes empirical assessment of methods in artificially simulated networks and realistic *in-silico* gene networks of high dimensionality. The paper concludes with Section 6, which summarizes the main findings and outlines directions for future work.

2. Background and Theory

In this section, general theory and background on causal modeling is provided.

2.1 Notation and Key Definitions

In this paper upper-case letters in italics denote random variables (e.g., A, B, C) and lower-case letters in italics denote their values (e.g., a, b, c). Upper-case bold letters in italics denote random variable sets (e.g., $\boldsymbol{X}$, $\boldsymbol{Y}$,$\boldsymbol{Z}$) and lower-case bold letters in italics denote their values (e.g., $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}$). The terms variables and vertices are used interchangeably. If a graph contains an edge $X \rightarrow Y$, then X is a parent of Y and Y is a child of X. An undirected edge X — Y denotes an adjacency relation between X and Y (i.e., presence of an edge directly connecting X and Y). A path p is a set of consecutive edges (independent of the direction) without visiting a vertex more than once. A directed path p from X to Y is a set of consecutive edges with same direction ("$\rightarrow$") connecting X with Y, i.e. $X \rightarrow ... \rightarrow Y$. X is an ancestor of Y (and Y is a descendant of X) if there exists a directed path p from X to Y. A directed cycle is a nonempty directed path that starts and ends on the same vertex X. We consider in this work two types of graphs: (i) directed graphs where vertices are connected only with edges "$\rightarrow$" and (ii) directed acyclic graphs (DAGs) without directed cycles and where vertices are connected only with edges "$\rightarrow$".

When the two sets of variables $\boldsymbol{X}$ and $\boldsymbol{Y}$ are conditionally independent given a set of variables $\boldsymbol{Z}$ in the joint probability distribution $\mathbb{P}$, we denote this as $\boldsymbol{X} \perp \boldsymbol{Y}|\boldsymbol{Z}$. For notational convenience, conditional dependence is defined as absence of conditional independence and denoted as $\boldsymbol{X} \not\perp \boldsymbol{Y}|\boldsymbol{Z}$. Two sets of variables $\boldsymbol{X}$ and $\boldsymbol{Y}$ are considered independent and denoted as $\boldsymbol{X} \perp \boldsymbol{Y}$, when X and Y are conditionally independent given an empty set of variables. Similarly, the dependence of X and Y is defined and denoted as $\boldsymbol{X} \not\perp \boldsymbol{Y}$.

We further refer the readers to (Pearl, 2009; Spirtes et al., 2000; Neapolitan, 2003; Glymour and Cooper, 1999) to review the standard definitions of conditional independence, collider, blocked path, d-separation, and causal sufficiency that are used in this work. Be-

low we review only several essential definitions:

Definition of local Markov condition: The joint probability distribution $\mathbb{P}$ over variables $\boldsymbol{V}$ satisfies the local Markov condition for a directed acyclic graph (DAG) $\mathbb{G} =< \boldsymbol{V}, \mathbb{E} >$ if and only if for each W in $\boldsymbol{V}$, W is conditionally independent of all variables in V excluding descendants of W given parents of W (Richardson and Spirtes, 1999).

Definition of global Markov condition: The joint probability distribution $\mathbb{P}$ over variables $\boldsymbol{V}$ satisfies the global Markov condition for a directed graph $\mathbb{G} =< \boldsymbol{V}, \mathbb{E} >$ if and only if for any three disjoint subsets of variables $\boldsymbol{X}$, $\boldsymbol{Y}$, $\boldsymbol{Z}$ from $\boldsymbol{V}$, if $\boldsymbol{X}$ is d-separated from $\boldsymbol{Y}$ given $\boldsymbol{Z}$ in $\mathbb{G}$ then $\boldsymbol{X}$ is independent of $\boldsymbol{Y}$ given $\boldsymbol{Z}$ in $\mathbb{P}$ (Richardson and Spirtes, 1999).

If the underlying graph $\mathbb{G}$ is a DAG, then the global Markov condition is equivalent to the local Markov condition (Richardson and Spirtes, 1999).

Definition of Bayesian network: $\mathbb{N} =< \mathbb{G}, \mathbb{P} >$ is a Bayesian network if the joint probability distribution $\mathbb{P}$ satisfies the local Markov condition for the DAG $\mathbb{G}$.

Next we provide an operational definition of causation and of a causal Bayesian network and local causal pathway. Notice that the following definition of causation matches the notion of randomized controlled experiment, which is the de facto standard for assessing macroscopic causation in the sciences (Pearl, 2009; Spirtes et al., 2000; Neapolitan, 2003; Glymour and Cooper, 1999).

Definition of causation, direct/indirect causation: Assume that a hypothetical experimenter can force a variable X to take specific values (i.e., to manipulate it). We say that X is a cause of Y (and Y is an effect of X) if the probability distribution of Y changes for some manipulation of X. X is the direct cause of Y with respect to $\boldsymbol{V}$, if: (i) X is a cause of Y, (ii) some manipulation of X would result in changes in the probability distribution of Y, no matter whether any variable in $\boldsymbol{V} \setminus \{X, Y\}$ were manipulated. If X is a direct cause of Y relative to $\boldsymbol{V}$, we say that there is a causal chain from X to Y. X is an indirect cause of Y with respect to $\boldsymbol{V}$ if there is a causal chain from X to Y of length greater than 2 (Pearl, 2009; Spirtes et al., 2000; Neapolitan, 2003; Glymour and Cooper, 1999).

We define causal Markov condition and causal Bayesian network by using the original definitions with the additional semantics that if there is an edge $A \rightarrow B$ in $\mathbb{G}$ then A directly causes B (for all A and $B \in \mathbb{V}$) (Spirtes et al., 2000).

Definition of local causal pathway: A local causal pathway of a target variable T is the set of its parents (direct causes) and children (direct effects) of T in the data-generative directed graph $\mathbb{G} =< \boldsymbol{V}, \mathbb{E} >$.

Definition of passenger: A passenger is a correlate of a target variable T and is neither a cause nor an effect of T.

Definition of local causal sufficiency: The variable set $\boldsymbol{V'}$ satisfies the local causal sufficiency condition if and only if it contains every common cause of all variables adjacent with a target variable T in the data-generative directed graph $\mathbb{G} =< \boldsymbol{V}, \mathbb{E} >$.

Next we provide several definitions of the faithfulness condition. This condition is essential for causal discovery from data.

Definition of graph faithfulness: If all and only the conditional independence relations that are true in $\mathbb{P}$ defined over variables $\boldsymbol{V}$ are entailed by the global Markov condition applied to a directed graph $\mathbb{G} =< \boldsymbol{V}, \mathbb{E} >$, then $\mathbb{P}$ and $\mathbb{G}$ are graph faithful to one another.

A relaxed version of graph faithfulness is given in the following definition:

Definition of adjacency faithfulness: Given a directed graph $\mathbb{G} =< \boldsymbol{V}, \mathbb{E} >$ and a joint

probability distribution P defined over variables $\boldsymbol{V}$, $\mathbb{P}$ and $\mathbb{G}$ are adjacency faithful to one another if every adjacency relation between X and Y in $\mathbb{G}$ implies that X and Y are conditionally dependent given every subset of $\boldsymbol{V} \setminus \{X, Y\}$ in $\mathbb{P}$ (Ramsey et al., 2006).

The adjacency faithfulness condition can be relaxed to focus on the specific target variable of interest:

Definition of local adjacency faithfulness: Given a directed graph $\mathbb{G} =< \boldsymbol{V}, \mathbb{E} >$ and a joint probability distribution $\mathbb{P}$ defined over variables $\boldsymbol{V}$, $\mathbb{P}$ and $\mathbb{G}$ are locally adjacency faithful with respect to T if every adjacency relation between T and X in $\mathbb{G}$ implies that T and X are conditionally dependent given any subset of $\boldsymbol{V} \setminus \{T, X\}$ in $\mathbb{P}$ (Statnikov et al., 2013).

It is known that some violations of the adjacency faithfulness condition can be attributed to violations of the intersection property of probability distributions (Pearl, 1997; Statnikov et al., 2013). This leads to distributions with variables that contain equivalent information (Statnikov et al., 2013; Lemeire, 2007). Such violations of the adjacency faithfulness condition constitute the focus of the paper because they are abundant in real biological networks, such as transcriptional gene regulatory networks (Statnikov et al., 2013; Statnikov and Aliferis, 2010; Dougherty and Brun, 2006), which are commonly investigated in computational causal discovery. For completeness, we also note that other violations of faithfulness exist in real biological networks and other real-life distributions, e.g. Simpsons paradox (Spirtes et al., 2000). While the latter violations may be equally important and not infrequent, they require a principally different treatment and often discovery techniques to address them are yet to be discovered; therefore we focus here only on violations due to information equivalencies.

Definition of target information equivalency: Two subsets of variables $\boldsymbol{X}$ and $\boldsymbol{Y}$ from $\boldsymbol{V}$ are target information equivalent with respect to a variable T iff the following conditions hold $T \not\perp \boldsymbol{X}$, $T \not\perp \boldsymbol{Y}$, $T \perp \boldsymbol{X}|\boldsymbol{Y}$, and $T \perp \boldsymbol{X}|\boldsymbol{Y}$ (Lemeire, 2007).

For example, consider a joint probability distribution $\mathbb{P}$ described by a causal Bayesian network with graph $A \rightarrow B \rightarrow T$ where A, B, and T are binary random variables that take values $\{0, 1\}$. Given the local Markov condition, the joint probability distribution can be defined as follows: $P(A = 0) = 0.3, P(B = 0|A = 1) = 1.0, P(B = 1|A = 0) = 1.0, P(T = 0|B = 1) = 0.2, P(T = 0|B = 0) = 0.4$. It follows that A and B contain equivalent information about T and adjacency faithfulness is violated because $T \perp \boldsymbol{B}|\boldsymbol{A}$.

While the above example showed information equivalencies resulting from deterministic relations, information equivalencies follow from a broader class of distributions with both deterministic and non-deterministic information equivalencies (e.g., see Figure 1 in Statnikov et al. (2013)).

Finally, we provide a definition of a near-faithfulness condition, which is going to be one of the sufficient assumptions for the novel causal discovery algorithms described in this work.

Definition of target information equivalency (TIE) near-faithfulness: A joint probability distribution $\mathbb{P}$ and a directed graph $\mathbb{G} =< \boldsymbol{V}, \mathbb{E} >$ are target information equivalency (TIE) near-faithful to one another if all violations of faithfulness can be attributed only to presence of target information equivalency relations in $\mathbb{P}$.

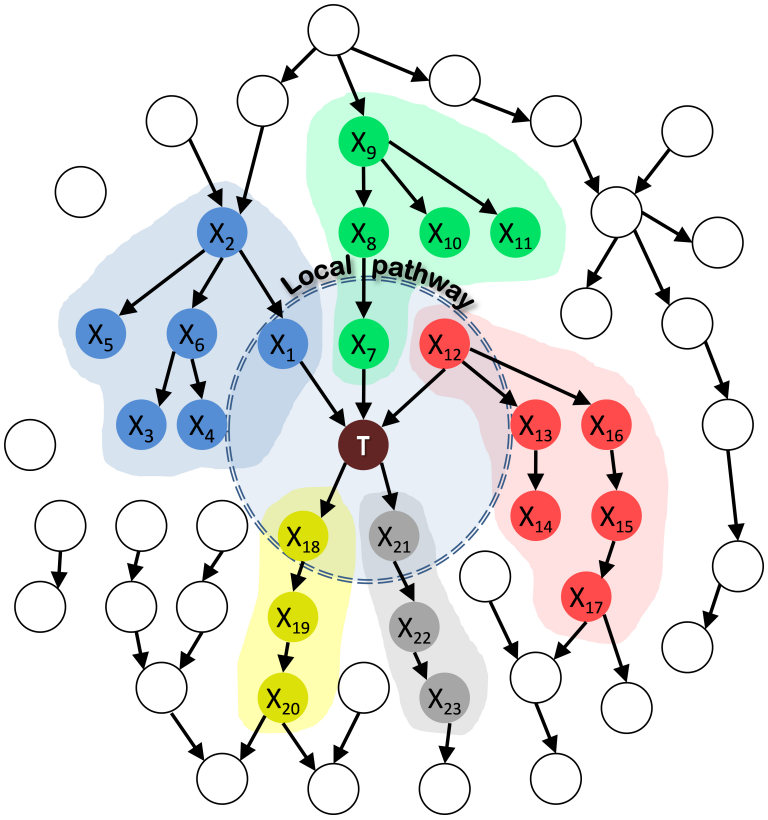


Figure 1: Graphical representation of an example TIE near-faithful causal network around a target variable T. The target variable T is shown in the middle of the network. Variables that are shown with the same color contain equivalent information about T. Variables in the local causal pathway of T are X_1, X_7, X_{12}, X_{18}, and X_{21}. Local causal discovery techniques that assume faithfulness (e.g., GLL-PC) will output one variable of each colored group. TIE* will output all subsets of the union of colored variables such that each subset has one variable from each colored group. No existing method will precisely determine the correct set $\{X_1, X_7, X_{12}, X_{18}, X_{21}\}$.

2.2 Local Causal Pathway Discovery from Observational Data in Faithful and Target Information Equivalency (TIE) Near-faithful Distributions

Prior research has provided sound conditional independence-based algorithms (e.g., GLL-PC from the Generalized Local Learning (GLL) family) for discovery of local causal pathway members from observational data under the assumptions of graph faithfulness (or local adjacency faithfulness with causal Markov condition), local causal sufficiency, and correctness of statistical decisions about dependence and independence (Aliferis et al., 2010a,b). To be precise, these methods only output the set of direct causes and effects of the target variable, but *do not distinguish which members of the output set are direct causes and which ones are direct effects*. The latter task requires randomized experiments or determination of edge

orientation through edge-orienting algorithms, temporal order, domain knowledge, or other post-processing criteria.

When the distribution is TIE near-faithful but not faithful, GLL-PC and other local causal discovery methods that assume faithfulness may lead to both false positives and false negatives in their output. Furthermore, false positives may neither be causes nor effects of the target variable. Consider an example causal network in Figure 1, which represents a TIE near-faithful distribution. The local causal pathway of the target variable T consists of five variables: $\{X_1, X_7, X_{12}, X_{18}, X_{21}\}$. Variables that are shown with circles of the same color contain equivalent information about T. For example, since X_1 and X_6 contain equivalent information about T, the following relation holds: $T \perp X_1 | X_6$. Thus, for example, GLL-PC may erroneously eliminate X_1 from the output (false negative) and conclude that X_6 is a member of the local causal pathway of T (false positive). In this distribution, there are 1,620 sets of five variables (= 6 'blue' × 5 'green' × 6 'red' × 3 'grey' × 3 'yellow' variables) that contain equivalent information about T. Notice that while only one of these 1,620 five-variable sets constitutes a local causal pathway of T, each of these five-variable sets can be arbitrarily output by GLL-PC or another local causal discovery algorithm that requires the same assumptions for soundness as GLL-PC, e.g. algorithms from (Peña et al., 2007). We say in such cases that there is a *multiplicity of local causal pathways consistent with the data*.

To address causal discovery in TIE near-faithful distributions, we have recently introduced two sound and complete algorithms TIE* and iTIE* (Statnikov et al., 2013) (described in Appendix F). These algorithms utilize conditional independence tests and allow discovery of all possible local causal pathways consistent with the data. In the example in Figure 1, these algorithms would identify all equivalency relations and output all 1,620 five-variable sets that span over variables $X_1, ..., X_{23}$. To further identify direct causes and direct effects of T in the variables output by the algorithms (the union of all equivalent sets of variables), one would need to resort to randomized experiments both because of target information equivalency and statistical indistinguishability of direct causes and effects in the context of local learning.

Now consider the global network learning methods such as SGS, PC (Spirtes et al., 2000), IC (Pearl, 2009), MMHC (Tsamardinos et al., 2006a), and LGL (Aliferis et al., 2010b) or region-based learning methods such as PCD-by-PCD (Yin et al., 2008), that under graph faithfulness, causal sufficiency, and correctness of statistical decisions can identify not only adjacency relations but also some edge orientations. The graphs output by these methods will be in general incomplete with regards to orientation because multiple graphs belong to the same Markov equivalence class of graphs and thus cannot be distinguished with observational data alone (Spirtes et al., 2000).

3. Prior Methods and Variants

Because learning a global causal network (that spans all measured variables) is substantially harder than learning a local causal pathway for a target variable, global methods fail to scale as the local ones. In order to experimentally test prior methods in high dimensional settings, we also introduce local versions of those that do not affect their soundness or quality.

Overall, we have considered 58 existing methods/variants spanning three main algorithmic families: conditional independence constraint-based structure learning (He and Geng, 2008; Meganck et al., 2006), linear cyclic models (Hyttinen et al., 2010; Eberhardt et al., 2010; Hyttinen et al., 2012) and Bayesian search-and-score (Pe'er et al., 2001; Sachs et al., 2005). These methods were chosen because they (i) reflect the current state-of-the-art in causal discovery, (ii) make use of observational and experimental data to produce directed causal networks, and (iii) are likely to scale to data of high dimensionality, unlike early methods for active learning of causal networks such as (Tong and Koller, 2001; Murphy, 2001). We describe the core ideas of each algorithmic family along with various variants below.

3.1 Conditional Independence Constraint-based Structure Learning

This family includes the ALCBN (Meganck et al., 2006) method and the method due to He and Geng (He and Geng, 2008). The main idea of these approaches is to learn an undirected[1] or partially directed graph from observational data (which represents the Markov equivalence class of graphs consistent with observational data), and then perform experiments to orient undirected edges. Both methods use the PC algorithm (Spirtes et al., 2000) to obtain an undirected or partially directed graph from observational data. The methods then use some decision criterion to select a variable for experimentation/manipulation, with the goal of maximizing the number of edges that are oriented after the experiment. The ALCBN algorithm uses either the mini-max, maxi-min or Laplace decision criteria (Meganck et al., 2006), whereas the method of He and Geng uses either the maxi-min or maximum entropy criteria (He and Geng, 2008). Once the variable is selected and manipulated, they perform a statistical independence test between the manipulated variable and each of its unoriented adjacencies in the graph, using experimental data. Adjacent variables that are associated with the manipulated variable are deduced to be direct effects, and all other adjacencies are direct causes (Spirtes et al., 2000). The ALCBN method repeats this process until all edges in the graph are oriented. The method of He and Geng first partitions the graph into chain components which are only connected by directed edges and orients each of these components separately. In addition to original methods, we also explored variants of these methods that restrict experimentation to the local causal pathway around a variable of interest/target or the chain component containing the variable of interest/target. A detailed list of all employed 24 methods/variants from this family (denoted as ALCBN and HE-GENG, accordingly) is given in Table A1 in Appendix A.

3.2 Linear Cyclic Models

This family includes three methods based on linear cyclic models with latent variables (Hyttinen et al., 2010; Eberhardt et al., 2010; Hyttinen et al., 2012). The main idea of these

1. The original methods considered using PC to learn a partially directed graph from observational data and then using experiments to further orient edges. Since orientation of PC is by design affected by errors in the adjacency structure, we also included in this work variants of these methods that work from the undirected graph obtained by PC from observational data.

approaches is to assume that all relations between variables are linear and can therefore be represented by an effects matrix. Discovering the causal structure then amounts to finding the coefficients of the effects matrix, which can be obtained by manipulating variables and deriving linear constraints on the effects. Specifically, these constraints are combined into a linear system and solved to obtain the coefficients of the effects matrix. Optionally, assuming faithfulness enables the use of the PC algorithm (Spirtes et al., 2000) on the manipulated and possibly observational data to learn adjacencies between variables. Non-adjacent variables imply additional constraints on the effects matrix, which are added to the linear system. The adjacencies also define an optimal order of variables to manipulate, which can minimize the number of required experiments. The derivation of constraints on the effects matrix and solution of the resulting linear system can be performed using any of the methods proposed by the authors, which we denote as LLC1 (Eberhardt et al., 2010), LLC2 (Hyttinen et al., 2010) and LLC3 (Hyttinen et al., 2012) ("LLC" stands for "linear, latents, cyclic"). The resulting effects matrix requires further filtering to obtain edges in the output graph. We used several approaches recommended by the authors: (i) removing all edges whose coefficients are less than a small fixed threshold, (ii) estimating the null distribution of the coefficients by rerunning the algorithm many times on permuted data and keeping only edges whose coefficients are statistically significant, and (iii) rerunning the algorithm a number of times on data sampled with replacement and keeping only edges whose mean coefficients are higher than their standard deviation. While the LLC method uses data for all variables in the network in order to estimate the effects matrix and produce the resulting causal graph, we limited the experiments only to variables with univariate association with the target (these methods have names beginning with "LLC"). In addition, we also experimented by limiting input data only for variables with univariate association with the target (these methods have names beginning with "UNIV-LLC"). A detailed list of all employed 32 methods/variants from this family is given in Table A2 in Appendix A.

3.3 Bayesian Search-and-score

This family includes the Biolearn method (Pe'er et al., 2001; Sachs et al., 2005). The main idea of this method is to define a space of candidate models, along with a scoring function that measures how well each model fits that data. Specifically, the score evaluates the posterior probability of a graph given the data. If given only observational data, graphs with the same undirected graph structure and unshielded colliders will have the same score (Neapolitan, 2003), and thus one can learn at best an equivalence class of graphs. Given experimental data, a score for each directed graph can be constructed by using the fact that the score decomposes into the local contributions of each variable. For each variable, only samples from experimental datasets where the variable was not manipulated were used, and the contributions of each variable were combined into a global score. This method can yield different scores for different orientations of the same graph structure, and thus can be used to evaluate how well directed graphs fit the combination of observed and manipulated data. Computing scores for all possible directed graphs is exponential in the number of variables, and thus it is usually not feasible to find the graph with the absolute highest score. Therefore, heuristics such as Greedy Hill-Climbing are used to limit the search space to a feasible

number. This method starts with an initial graph structure (such as the empty graph) and computes the score for closely related graphs obtained by adding, removing or reversing different edges. It selects the graph with the highest score, and repeats the procedure until it has found a local maximum. The entire process is repeated many times (e.g., 500), and the final model consists of all the edges present in a significant portion (85%) of the graphs. We used two variants of this method: one with the Normal Gamma scoring function (denoted as BIOLEARN.NG) and another one with the BDE scoring function (denoted as BIOLEARN.BDE).

4. New Methods

Below we provide new algorithms for local causal discovery. These algorithms rely on observational data for identifying members of the local causal pathway of a target variable; *however all orientation decisions are based on experimental data exclusively.* While prior research has provided theoretically sound approaches for orienting edges from observational data (e.g., V-structure based orientation in PC algorithm (Spirtes et al., 2000)), the empirical accuracy of these methods is affected by errors in constructing undirected skeleton and violations of faithfulness. We provide in Appendix D and Table D1 an empirical comparison of orientation approaches that concludes that significantly higher quality of orientation can be achieved from experimental data.

4.1 Algorithm ODLP*

In order to facilitate comprehension of the general methodology, we first address the problem of local causal pathway discovery in faithful distributions. The algorithm ODLP* is shown in Figure 2.

Theoretical analysis of the algorithm correctness: ODLP* is sound and complete under the sufficient assumptions of (i) local adjacency faithfulness; (ii) causal Markov condition; (iii) local causal sufficiency; (iv) acyclicity of the data-generative graph; and (v) correctness of statistical decisions. The proof of correctness relies on a previously established theoretical result showing that GLL-PC algorithms can identify members of the local causal pathway (direct causes and direct effects of the target variable) from observational data under the above stated assumptions (Aliferis et al., 2010a,b). In principle ODLP* can call another sound and complete algorithm for identification of local causal pathway members in step 1. Notice however, that algorithms for identification of local causal pathway members (such as GLL-PC) do not differentiate between direct causes and direct effects in the local causal pathway, and in general this task has to be accomplished with additional experimental data, as outlined in steps 2 and 3 of ODLP*.

Trace of the ODLP* algorithm: Consider running the ODLP* algorithm on observational data generated from the causal graph shown in Figure 2. We aim to identify the local causal pathway of the target variable T. In step 1 of ODLP*, GLL-PC will identify that variables X_1, X_2, X_3, X_4, X_5 belong to the local causal pathway of T, however would not define causal role of any of these variables. If it is possible to manipulate T, we would do so (step 2.a) and reveal that X_4 and X_5 change due to manipulation of T, and thus are direct effects of T (step 2.b); the remaining variables $X_1, X_2, and X_3$ thus have to be direct

Algorithm ***ODLP****

- *Input*:
  - Observational data D^O, including a target variable *T*;
  - Experimental protocols/methods to manipulate one variable at a time and generate experimental data D^E that quantifies response of the system to the manipulation.
- *Output*: Local causal pathway of *T*.

1. Apply *GLL-PC* or another sound and complete method to the observational data D^O to identify the set of variables ***V*** that are members of the local causal pathway of *T*.
2. If it is possible to manipulate *T*,
   a. Manipulate *T* and obtain experimental data D^E.
   b. Mark all variables in ***V*** that change in D^E due to manipulation of *T* as "*direct effects*" and mark remaining variables in ***V*** as "*direct causes*".
3. Else
   a. Manipulate a variable *X* in ***V*** to obtain experimental data D^E.
   b. If *T* changes in D^E due to manipulation of *X*, mark *X* as a "*direct cause*"; otherwise mark *X* as a "*direct effect*".
   c. Repeat steps 3.a and 3.b for all variables in ***V***.
4. Return the local causal pathway of *T*.

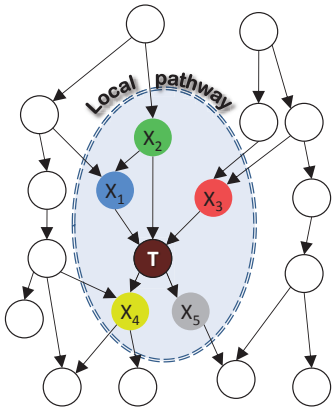


Figure 2: Pseudo-code of the ODLP* algorithm for faithful distributions. Left: Pseudo-code of the algorithm. Right: Graphical representation of an example causal network around a target variable T. Variables are shown with white circles, and edges represent direct causal influences. Variables in the local causal pathway of T are X_1, X_2, X_3, X_4, and X_5.

causes of T (step 2.b). On the other hand, if T cannot be manipulated, we can manipulate X_1 (step 3.a) and observe that T changes due to manipulation of X_1 (step 3.b); therefore X_1 is a direct cause of T (step 3.b). If we consider manipulating X_4 (step 3.a), we would observe that T does not change due to manipulation of X_4 (step 3.b); therefore X_4 is a direct effect of T (step 3.b). When steps 3.a and 3.b are applied to other variables in the local causal pathway, we will also find two additional direct causes of T (X_2 and X_3) and one additional direct effect (X_5) of T.

Analysis of the algorithm's experimental strategy and its efficiency: The experimental strategy of ODLP* is efficient because it relies only on single-variable manipulation experiments that are expected to generate a small number of samples in order to assess univariate association of the manipulated variable with all other variables. Furthermore, the algorithm tries to minimize the number of single-variable manipulation experiments and will conduct only one experiment if T can be manipulated (step 2.a). If it is not possible to manipulate T (e.g., T is a disease in humans), it will conduct the same number of experiments as the number of variables in the output of GLL-PC (set $\boldsymbol{V}$). In the most general case, it is impossible to further reduce this number of experiments because every variable in $\boldsymbol{V}$ can potentially be a direct cause of T and has to be confirmed by an experiment. We

note that situations exist that can lead to additional savings in experiments (e.g., when X, a direct effect of T, is causing Y, another direct effect of T, then manipulation of X would also reveal that Y is an effect of T and save an experiment) and we do check for them in the algorithm implementation, although they are not described in the algorithm pseudo-code in order to help understanding of its basic principles. Finally, it is also worthwhile to point out that the ODLP* algorithm can incorporate background knowledge both during the stage of learning the local causal pathway members (step 1) and when determining the causal role of the involved variables (steps 2 and 3), which can potentially lead to further reducing the number of required manipulation experiments.

We note that ODLP* does not represent a radical departure over previously known algorithms (it is a modest extension of preexisting ideas), however it is essential to conceptually describe the much more complex and generally applicable algorithm ODLP.

4.2 Algorithm ODLP

A more general algorithm ODLP shown in Figure 3 addresses the problem of local causal pathway discovery in TIE near-faithful distributions. This algorithm is specifically designed for situations when the target variable T can be manipulated.

Theoretical analysis of the algorithm correctness: The following theorem states correctness of ODLP; the proof is given in Appendix G. Specifically, the theorem shows that under certain assumptions, ODLP will return all and only members of the true local causal pathway of a target variable T.

Theorem 1 *ODLP is sound under the following sufficient assumptions: (i) TIE near-faithfulness (as a relaxation of local adjacency faithfulness to allow for target information equivalency relations); (ii) causal Markov condition; (iii) local causal sufficiency; (iv) acyclicity of the data-generative graph; and (v) correctness of statistical decisions.*

In non-technical terms, the first two assumptions mean that with the exception of empirical target information equivalency relations, there is a direct correspondence between the data and a directed acyclic data-generative graph in terms of statistical relations (specifically, there is an edge between two variables if and only if they have association in the data conditioned on every subset of other variables). The third assumption means that every common cause of two or more measured variables is also measured in the dataset. If this assumption is violated, direct causation cannot be discovered by using observational data together with experiments limited to single-variable manipulations, as demonstrated in Figure 1 of (Eberhardt et al., 2010). The fourth assumption means that there are no feedback cycles in the graph. The fifth assumption means that determination of variable (in)dependency in the population from the available data sample is correct.

Trace of the ODLP algorithm: Consider running ODLP on data generated from the network in Figure 1. The algorithm aims to identify the local causal pathway of the target variable T. In step 1, TIE* will find 1,620 local causal pathways of T consistent with the data. The union of these data-consistent pathways (set $\boldsymbol{V}$) will be variables $X_1, ..., X_{23}$ (step 2). Then in step 3, ODLP will form five equivalence clusters of variables based on

Algorithm ***ODLP***

- *Input*:
  - Observational data D^O, including a target variable *T*;
  - Experimental protocols/methods to manipulate one variable at a time and generate experimental data D^E that quantifies response of the system to the manipulation.
- *Output*: Local causal pathway of *T*.

1. Apply TIE* or iTIE* to the observational data D^O to identify all local causal pathways of *T* consistent with the data.
2. ***V*** ← Union of all variables that participate in local causal pathways of *T* consistent with the data (*this is a draft of the local causal pathway*).
3. Form equivalence clusters over variables in ***V*** such that each equivalence cluster contains variables that have equivalent information about T (*this can be accomplished directly from the output or the operation of TIE* or iTIE**).

Identify effects of T

4. Manipulate *T* and obtain experimental data D^E.
5. Mark all variables in ***V*** that change in D^E due to manipulation of T as "*effects*".

Identify direct and other causes of T

6. Repeat
   a. If there is an equivalence cluster that contains a single unmarked variable *X* and all marked variables in this cluster (if any) are only passengers and/or effects, then mark *X* as a "*direct cause*" and go to step 6.
   b. Select (according to some heuristic function or at random) an unmarked variable *X* from an equivalence cluster.
   c. Manipulate *X* and obtain experimental data D^E.
   d. If *T* does not change in D^E due to manipulation of *X*, mark *X* as a "*passenger*" and mark all other non-effect variables that change in D^E due to manipulation of *X* as "*passengers*"; otherwise mark *X* as a "*cause*".
7. Until there are no equivalence clusters with unmarked variables.
8. For every cause *X*, mark *X* as a "*direct cause*" if there exist no other cause in the same equivalence cluster that changes due to manipulation of *X*; otherwise mark *X* as an "*other cause*".

Identify direct effects of T

9. Repeat
   a. If there is an equivalence cluster that contains a single effect variable *X* which has neither been marked as "other effect" nor as "direct effect" and other effect variables in this cluster (if any) are only other effects, then mark *X* as a "*direct effect*" and to go step 9.
   b. Select (according to some heuristic function or at random) an effect variable *X* that has neither been marked as "*other effect*" nor as "*direct effect*".
   c. Manipulate *X* and obtain experimental data D^E.
   d. Mark all effect variables that change in D^E due to manipulation of *X* and belong to the same equivalence cluster as "other *effects*".
10. Until all effect variables are either marked as "*other effects*" or *"direct effects"*.
11. Return the local causal pathway of *T*, i.e. only direct causes and direct effects of *T*.

Figure 3: Pseudo-code of the ODLP algorithm for TIE near-faithful distributions. Notice that even though the algorithm outputs the local causal pathway of T, during its execution it also discovers the causal role of other variables that will provide additional clues about underlying mechanisms. Steps 4, 6.c, 9.c provide an interface of the algorithm with the external world through experiments that are conducted by an experimentalist, and are shown with dark grey highlighting.

information that they provide about the T (the clustering will coincide with the color of highlighting of variables in Figure 1). In steps 4 and 5 the algorithm will manipulate T and identify its effects $X_{18}, ..., X_{23}$. Then the algorithm will proceed to identification of direct/other causes ("other causes" are defined as causes that are not identified as direct causes, they could be indirect causes or both direct and indirect causes at the same time) of T in the candidate set of variables $X_1, ..., X_{17}$. There is no equivalence cluster that satisfies criterion of step 6.a, so ODLP will proceed to step 6.b and select a variable for manipulation (for example without loss of generality, X_6) in step 6.c. The algorithm will then identify that X_6 is a passenger and so are X_3 and X_4 (step 6.d). Steps 6.a-6.d will be repeated until the causal role of every non-effect variable is deciphered. Next, the algorithm will conclude that X_1, X_7, and X_12 are direct causes of T and X_2, X_8, and X_9 are other causes (step 8). Then ODLP will proceed to the identification of direct effects and other effects of T in the set of effects $(X_{18}, ..., X_{23})$. Similarly, "other effects" are effects that are not identified as direct effects, they could be indirect effects or both direct and indirect effects at the same time. There is no equivalence cluster that satisfies criterion of step 9.a, so the algorithm will proceed to step 9.b and select a variable for manipulation (for example without loss of generality, X_19) in step 9.c. In step 9.d ODLP will identify that X_20 is other effect of T and repeat iterations until all effects are either marked as "other effects" (X_{19}, X_{20}, X_{22}, and X_{23}) or "direct effects" (X_{18} and X_{21}). Thus the local causal pathway of T (that consists of direct causes X_1, X_7, X_{12} and direct effects X_{18}, X_{21}) has been identified correctly.

Analysis of the algorithm's experimental strategy and its efficiency: The strategy of ODLP relies on single-variable manipulation experiments and usually requires a small number of samples from each experiment to assess univariate associations of the manipulated variable with other variables. In general, the number of experiments necessary for identification of the local causal pathway would be manageable for experimentalists, although it varies and depends on the structure of the local causal pathway. The number of experiments for the best and worst case is 1 and $|\boldsymbol{V}| + 1$, respectively, where the set V is the union of all variables that participate in local causal pathways of T consistent with the data. In any case, the number of experiments would be manageable because $\boldsymbol{V}$ in real distributions, even in high-throughput datasets, is typically between 10 and 200 variables, as we have observed by running TIE* in > 30 datasets (Statnikov et al., 2013; Statnikov and Aliferis, 2010).

An important principle behind minimization of experiments is to first manipulate in step 6.c passengers of T that are causing many other passengers of T. For example, manipulation of X_6 in Figure 1 would lead to changes in X_3, X_4 but not in T. Therefore, X_3, X_4, and X_6 are not causes of T. The algorithm can also infer from manipulation of T that X_3, X_4, and X_6 do not change and thus are not effects of T. Therefore, they are passengers. The algorithm determined the causal role of X_3, X_4, and X_6 by manipulating only one of these variables. However, the graphical structure is not known when the algorithm performs experiments, and thus it has to resort to heuristics to manipulate first variables that are likely to yield savings in experiments. The algorithm uses a partial network-based heuristic that chooses a variable that has the highest topological order relative to T. The topological order can be established from constraints learned from experimental data, as well as from domain knowledge, temporal order information, computational edge orientation algorithms based on observational data, and other sources. In addition to the above

Network Name	Description	Construction Methodology	Num. Variable	Num. Edges	Num. Samples in Obs. Data	Num. Samples in Exp. Data	Reference
REGED	Resimulated transcriptional gene regulatory network from gene expression data of human lung cancer patients. Variables represent expression levels of genes, and target variable represents lung cancer subtype.	Used a publicly available microarray gene expression dataset to learn a network structure of transcriptional interactions. Parameterized the network using non-linear regression.	1,000	1,148	500	100	Guyon et al. (2008)
ECOLI	Resimulated transcriptional gene regulatory network based on the current knowledge of regulation in E.Coli. Variables represent expression levels of genes.	Used large-scale experimental data to infer the network structure, and then used principles of thermodynamics and molecular kinetics to parameterize the network.	1,565	3,648	1000	200	Marbach et al. (2009); Schaffter et al. (2011)
YEAST	Resimulated transcriptional gene regulatory network based on the current knowledge of regulation in S. Cerevisiae. Variables represent expression levels of genes.	Same as above	4,441	12,873	1000	200	Marbach et al. (2009); Schaffter et al. (2011)
P1000	Artificially simulated network, where the target information equivalency phenomenon is present in the local causal pathway of the target variable. As a result, the target variable has multiple 1,620 data-consistent local causal pathways.	Manually generated graph of the network and parameterized using Gaussian distribution.	1,000	51	1000	20	Novel
P1M	Large-scale version of P1000 network with 1,000,000 variables.	Tiled with P1000 as the basic component with inter-tile connections.	1,000,000	81,969	1000	20	Novel

Table 1: Description of networks and data used in empirical experiments.

heuristic, other heuristic functions can be used. We refer interested readers to Appendix H for more detailed examples explaining ODLP's experimental strategy and its efficiency. Similarly, prioritizing manipulation of direct effects in step 9.c allows saving experiments by avoiding manipulation of indirect effects. Finally, it is also worthwhile to point out that the ODLP algorithm can incorporate background knowledge both during the stage of drafting the local causal pathway (step 1) and when determining the causal role of variables (steps 4-10), which can potentially lead to further reducing the number of required experiments.

We also note that in settings when the assumptions of the algorithm are violated and TIE* outputs false positives, one may choose not to perform step 6.a and always manipulate a single unmarked variable in the equivalence cluster to ensure that it is indeed the cause. Otherwise, a false positive variable (e.g., passenger in the equivalence cluster consisting of one variable) will be erroneously classified as "direct cause" in step 6.a. However, when the sufficient assumptions of the algorithm hold, step 6.a does not lead to errors and provides savings in the number of experiments. Similarly, step 9.a can be omitted which leads to improving robustness in handling false positives but decreasing experimental efficiency.

5. Empirical Experiments

This section describes the data used in the empirical experiments, implementation of different causal discovery algorithms, performance metric and statistical comparison methods, and results of the empirical experiments.

5.1 Networks and Data

The networks and data used in empirical experiments are summarized in Table 1. The REGED, ECOLI, and YEAST networks produce resimulated gene expression data that resembles data from real transcriptional gene regulatory networks. Since these networks have been previously published (Guyon et al., 2008; Marbach et al., 2009; Schaffter et al., 2011), we do not describe them in detail here. We will only mention that variables in ECOLI and YEAST networks (genes) typically have very few (0-2) direct causes (direct upstream regulators), and some variables (transcription factor genes) have a large number of direct effects (direct downstream targets) that can even reach low hundreds. This is consistent with the principles of transcriptional regulation. The P1000 network is intended i) to resemble data from real transcriptional gene regulatory networks which are generally very sparse and ii) to demonstrate the effect of multiplicity of causal pathways consistent with the data, a phenomenon which is omni-present in real biological networks (Statnikov et al., 2013; Statnikov and Aliferis, 2010; Dougherty and Brun, 2006). This network was obtained by parameterizing the local causal pathway structure shown in Figure 1 using linear Gaussian distribution and adding unconnected Gaussian variables, so that the total number of variables is 1,000. The parameterization of the network is provided in Table B1 in Appendix B. The P1M network was obtained by "tiling" the P1000 network one thousand times. The structural and probabilistic properties of individual tiles were similar to that of P1000, so that the distribution of P1M network resembles the distribution of the P1000 network. More specifically, one thousand copies (i.e. tiles) of the P1000 network were generated, each copy with the set of vertices $\boldsymbol{V_i}$ and the set of edges $\boldsymbol{E_i}$ that are copies of

$\boldsymbol{V_{P1000}}$ and $\boldsymbol{E_{P1000}}$, the vertex set and the edge set of the original P1000 network. Then, the tiles were interconnected with edges between $\boldsymbol{V_i}$ to $\boldsymbol{V_{i+1}}$. The vertices that received inter-tile edges were re-parameterized to preserve their marginal distribution, following the approach described in (Tsamardinos et al., 2006b). See Figure B1 in Appendix B for visualization of the fragment of the connected components of the P1M network.

We generated 1,000 samples for the observational datasets for all networks, except for REGED because this network has been previously used with 500 samples in the international challenge on Causation and Prediction (Guyon et al., 2008). Prior to running experiments, we generated experimental datasets by manipulating each variable in every network. The sample size for experimental datasets was minimized for each network over $\{20, 100, 200\}$ in order to be realistic and at least have sufficient power to estimate univariate associations of manipulated variables with other variables in the network. As a result, we used 100 samples in REGED, 200 in ECOLI and YEAST, and 20 in P1000 and P1M networks for experimental datasets. All generated experimental datasets were saved in a working database. Causal pathway discovery methods could query this database to obtain an experimental dataset where the variable of interest was manipulated. The decoupling of the two most time consuming components of experiments, simulation of experimental data and running causal discovery algorithms, allowed us to setup a robust algorithm evaluation environment (Figure 4). All data for the simulations is available on the manuscript supplemental website: `http://ccdlab.org/odlp.html`.

Since we are focusing here on discovery of local causal pathways, the next step is to select target variables of interest. The networks REGED, P1000, and P1M have designated target variables. However, there are no designated targets in YEAST and ECOLI networks. Therefore, we selected four variables from each network (the number of selected variables was limited by computational resources of the study) and designated them as targets. These four variables were selected randomly from the subset of transcription factors (that play key regulatory role in these networks) such that they represent local causal pathways of varying sizes for each network. This also allows assessing sensitivity of methods to the size of the local causal pathway. More details are given in Table 2.

5.2 Local Causal Pathway Discovery Methods and Implementations

In addition to ODLP, we evaluated 58 existing methods/variants for active learning of causal networks that are described in Section 3. ODLP and conditional independence constraint-based structure learning methods ALCBN and HE-GENG were implemented in Matlab and used the implementation of Fisher's Z test of conditional independence from the Causal Explorer library (Statnikov et al., 2010). ODLP was run using the iTIE* algorithm to find all data-consistent local causal pathways, parameter max-k (denoting maximum cardinality of the conditional test) set to 3, and 0.05 alpha for assessing dependence/independence. ALCBN and HE-GENG used the implementation of the PC algorithm from the Causal Explorer library (Statnikov et al., 2010) and were run with maximum cardinality of conditional tests set to 2 and 0.05 alpha for assessing dependence/independence. We tried to run the algorithms with larger cardinality of conditional tests, but it was not computationally feasible because PC did not terminate in most cases in less than one month of single-core

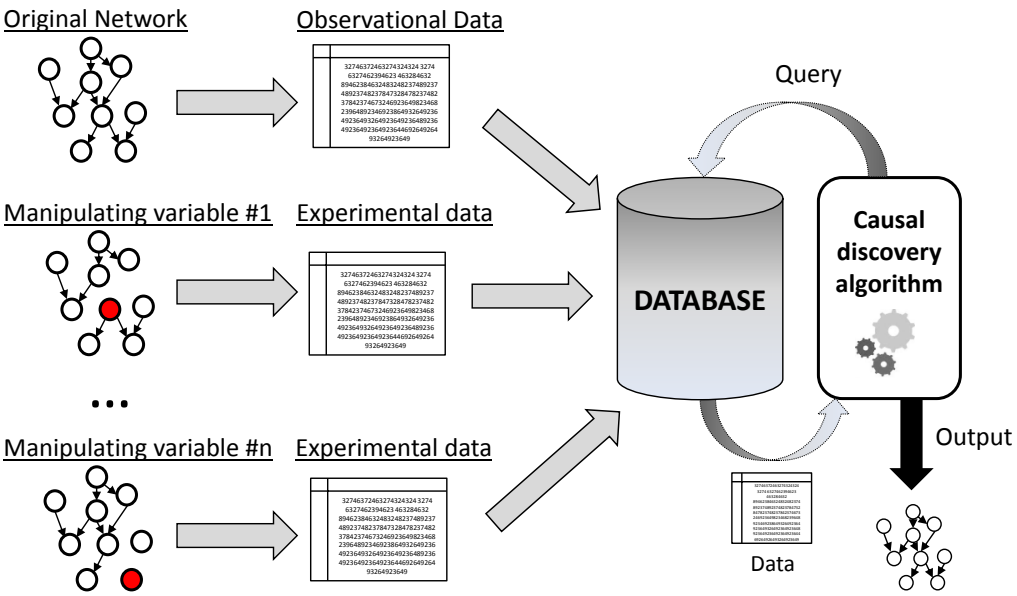


Figure 4: Data generation process/experimental setup. The depicted experimental setup allowed to decouple data generation and running of algorithms, therefore providing a robust algorithm evaluation environment.

time. We used the original authors' R implementations of methods based on linear cyclic models (obtained directly from the authors) and improved their efficiency in Matlab, e.g. to solve very large-dimensional sparse linear systems that cannot be solved easily in R due to current memory restrictions. Finally, we used the original authors' software implementation of the Bayesian search-and-score method. Table E1 in Appendix E provides information and location of publicly available software implementations of the above discovery methods.

5.3 Performance Metrics and Statistical Comparison

Assessment of the performance of algorithms was based on the following metrics: (i) sensitivity, (ii) specificity, and (iii) number of required experiments. Sensitivity and specificity are metrics to assess the accuracy of structure learning, and they were computed for the task of discovery of all direct causes and all direct effects of the target variable T. Sensitivity and specificity range from 0 to 1 (or 0% to 100%), with larger values denoting better performance. We also combined sensitivity and specificity into a single metric, the Euclidean distance from the point with sensitivity and specificity equal to 1: $\frac{\sqrt{(1-sensitivity)^2+(1-specificity)^2}}{\sqrt{2}}$. The latter metric is referred to as "distance" in the manuscript and it ranges from 0 to 1 (or 0% to 100%), with larger values denoting worse performance. In addition to using the raw values for the number of experiments, we also normalized this metric by dividing it by the number of variables in the local causal path-

Network Name	Target Variable T	Num. Variables in the Local Causal Pathway of T	Num. Direct Causes of T	Num. Direct Effects of T
REGED	Adenocarcinoma vs. squamous lung cancer subtype.	15	2	13
ECOLI	Expression levels of gene agaR	8	0	8
	Expression levels of gene allR	10	0	10
	Expression levels of gene zur	6	0	6
	Expression levels of gene lexA	54	0	54
YEAST	Expression levels of gene YBL005W	30	1	29
	Expression levels of gene YFL044C	15	0	15
	Expression levels of gene YLR014C	31	0	31
	Expression levels of gene YKL112W	300	2	298
P1000	Artificial	5	3	2
P1M	Artificial	5	3	2

Table 2: Description of target variables chosen from each network and their local causal pathways. As mentioned in the manuscript, the small number of direct causes of the target variables in ECOLI and YEAST networks is representative of these two networks and principles of transcriptional regulation.

way of T or by the number of variables in the entire network. To test whether the differences in distance metric between the nominally best performing algorithm and other algorithms are non-random for a specific local causal pathway discovery task, we used a statistical permutation-based test adapted from (Menke and Martinez, 2004). We obtained a null distribution for each comparison task and computed the corresponding p-value. When the p-values are not statistically significant at 0.05 alpha level after adjusting for multiple comparisons using the methodology of (Benjamini and Hochberg, 1995; Benjamini and Yekutieli, 2001), the resulting algorithms are deemed to have statistically comparable distance with the algorithm with best distance value. We refer to such values of distance metric as 'optimal' for a specific local causal pathway discovery task relative to the tested methods.

5.4 Computing Resources and Execution of Experiments

To run the experiments, we used three high-performance computing (HPC) clusters available to us at the time of experiments. These HPC clusters included: the Asclepius cluster of the NYU Langone Medical Center, the Bowery cluster of the New York University main campus, and the BuTina cluster of the New York University Abu Dhabi campus in the United Arab Emirates. Asclepius had $\sim 1,000$ Intel x86 processing cores and 4TB of RAM distributed among the cluster's compute nodes. The Bowery cluster had $\sim 2,500$ cores and

9TB of RAM total among all the nodes. The BuTina cluster had $\sim 6,400$ latest Intel x86 processing cores with a total of 26TB of RAM.

In addition to distributing the tasks of running various causal pathway discovery algorithms for various networks/target variables among compute cores of the cluster, we also often divided the individual tasks (of running a single algorithm for a specific network/target) into many sub-tasks. For example, Biolearn requires running Greedy Hill-Climbing procedure 500 times, all of which can be run independently on individual cores. In many cases, the independent nature of the sub-tasks enabled linear speedup. In order to complete execution of experiments with available resources, we imposed three termination criteria: (i) 30 day single-core time limit for tasks that cannot be easily parallelized; (ii) 3,000 day multi-core time limit for tasks that can be further parallelized (spread over 100 cores); and (iii) 48 GB RAM. We used 500-700 cores at a time over 2.5 calendar years. We estimate that the final results reported here required 800 core-years of computation.

5.5 Results

The detailed results of experiments are provided in Table C1 (for REGED, P1000, and P1M networks), Table C2 (for ECOLI network), and Table C3 (for YEAST networks) in Appendix C. These tables provide values of sensitivity, specificity, distance, and number of experiments for each method and local causal pathway discovery task. As mentioned in the previous sub-section, in some cases experiments were terminated due to extensive computational resource requirements or, for Biolearn, failure of the original software implementation of the method. These cases are marked in the tables with special codes T1, T2, T3, T4, and the legend is given in Table C1.

Before reporting detailed analysis of the results, it is worth noting that the ODLP algorithm resulted in better performance than any other algorithm when applied to the P1000 dataset. This is partly due to the fact that TIE* and the ODLP algorithm specifically address local pathway multiplicity, which is present in P1000 dataset. On the other hand, many other algorithms rely on the PC algorithm, which assumes faithfulness.

Analysis based on the counts of successes/failures: In the following analyses, presented in Figures 5-8, we provide for each method the number of counts of successes/failures (according to various metrics) within 11 local causal discovery pathway tasks.

Figure 5 reports for each method the number of local causal pathway discovery tasks where a method either exceeded available computational resources or its original software implementation failed to run. ODLP is the only method that was able to run for all 11 local causal pathway discovery tasks. No other method was able to run for P1M network with 1,000,000 variables. However, within each algorithmic family except for Biolearn, there are methods that were able to run on the remaining 10 local causal pathway discovery tasks (represented by a failure number of 1). From ALCBN and HE-GENG families, these are mostly methods restricted to the local neighborhood of the target variable. From LLC family, these are methods that use only variables with significant univariate association with the target variable. This observation motivates the approach of using local methods for solving local causal pathway discovery problems. Also, for ALCBN and HE-GENG methods that discover the global network, the ones that use undirected PC skeleton (ALCBN.S. or

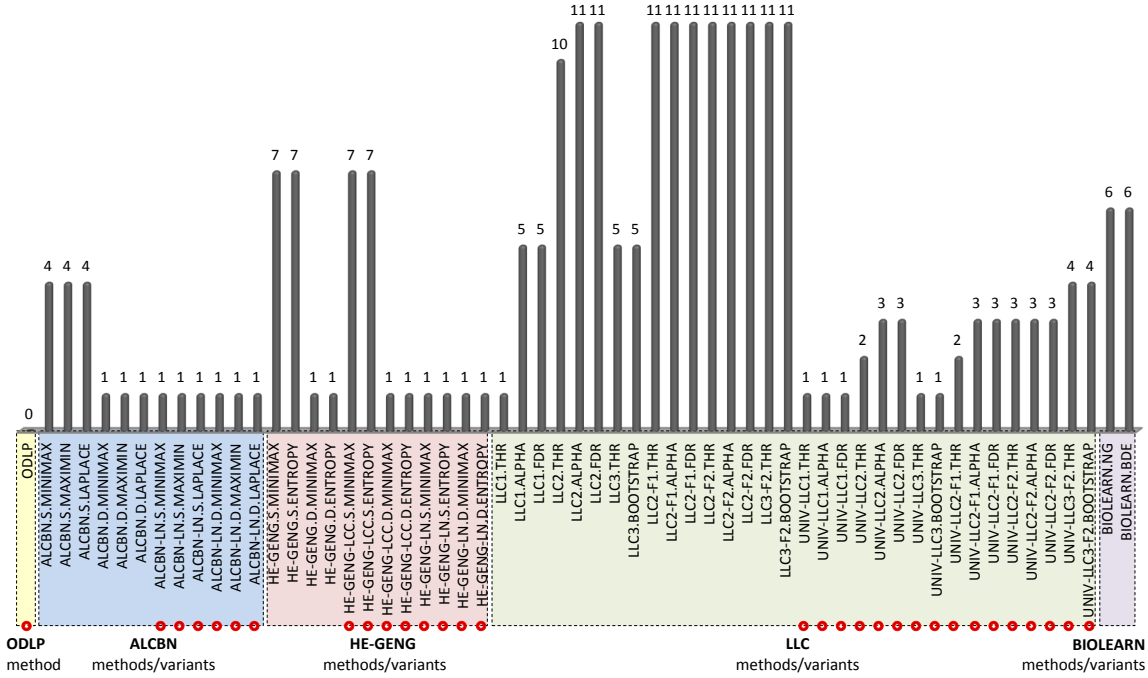


Figure 5: Number of local causal pathways where the algorithm was terminated/failed (out of 11 local causal pathways). Red circles denote methods designed for the discovery of local causal pathways. These include our modifications of the original global methods for local learning.

HE-GENG.S.*) fail more often in comparison to the ones that use partially directed global graph (ALCBN.D.* or HE-GENG.D.*). This is due to the fact that more computation is needed to determine which variable(s) to manipulate in the completely undirected graph, and our experiments have a restriction on computational time. It is also worthwhile to mention that the runtime of ODLP was under 10-15 minutes for all pathways, except for YEAST pathway for gene YKL112W where it took the algorithm one hour to run because the underlying local causal pathway was of large size (300 members). Other methods took orders of magnitude more computing time, e.g. it took ALCBN and HE-GENG of the order of 10 hours to obtain unoriented PC skeleton, and it took LLC of the order of several days to derive constraints on the effects matrix and combine them into a linear system. These run-time estimates are for the major computing components of the core methods, without bootstrapping/permutations. If the latter techniques are used, the run-time typically increases by more than two orders of magnitude due to a large number of independent runs of the core method.

Figure 6 reports for each method the number of local causal pathway discovery tasks where a method achieved optimal value of the distance metric (defined as a distance value that is not significantly different from the best distance achieved by all method examined,

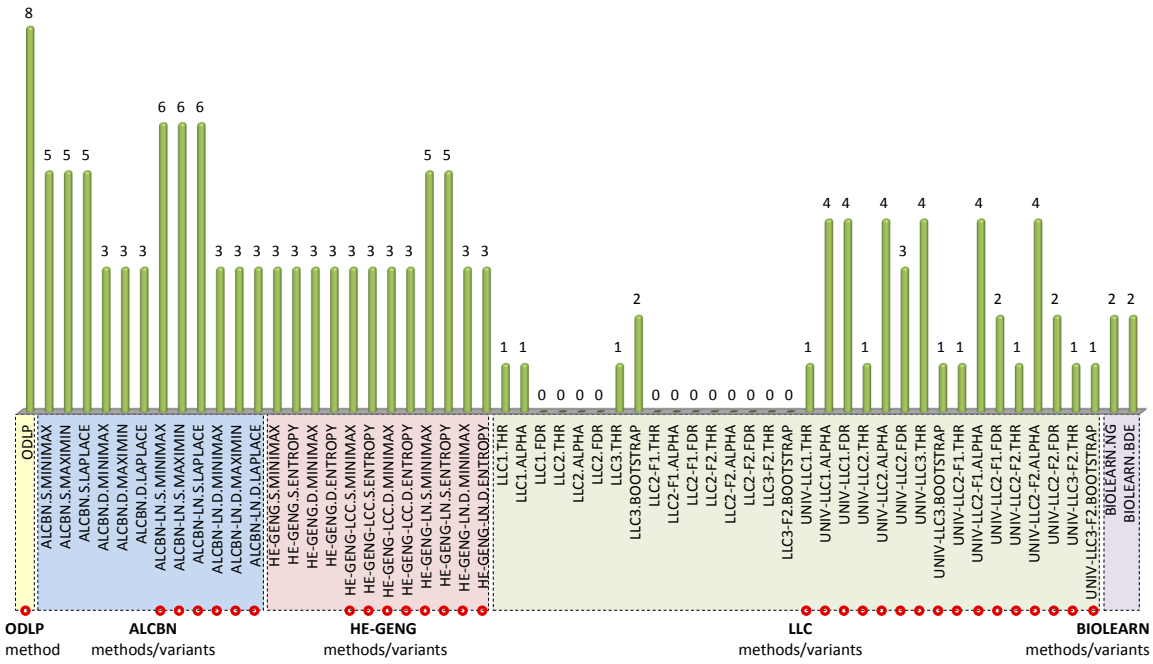


Figure 6: Number of local causal pathways discovered by an algorithm with optimal distance (out of 11 local causal pathways).Red circles denote methods designed for the discovery of local causal pathways. These include our modifications of the original global methods for local learning.

reflecting accuracy of structural discovery of the pathway). ODLP achieved optimal distance in eight out of 11 pathways, local versions of ALCBN based on unoriented PC skeleton achieved optimal distance in six pathways, local versions of HE-GENG based on unoriented PC skeleton and versions of ALCBN based on unoriented skeleton achieved optimal distance in five pathways, and some versions of LLC limited to variables univariately associated with the target achieved optimal distance in four pathways. Other methods achieved optimal distance in three or fewer pathways.

Figure 7 reports for each method the number of local causal pathway discovery tasks where a method achieved optimal values of the distance metric and did not perform more experiments than the number of members in the pathway. Figure 8 provides similar data but for the number of experiments limited by 10, which is commonly used in biological sciences for expensive experiments. In both analyses, ODLP and local versions of ALCBN based on the unoriented PC skeleton succeeded in six out of 11 pathways. Local versions of HE-GENG also based on the unoriented PC skeleton succeed in five (if the number of experiments is limited by the number of members in the pathway) or four (if the number of experiments is limited by 10) pathways. Two versions of LLC limited to variables univariately associated with the target succeeded in four pathways (if the number of experiments is limited by the number of members in the pathway). All other methods/variants succeeded in three or

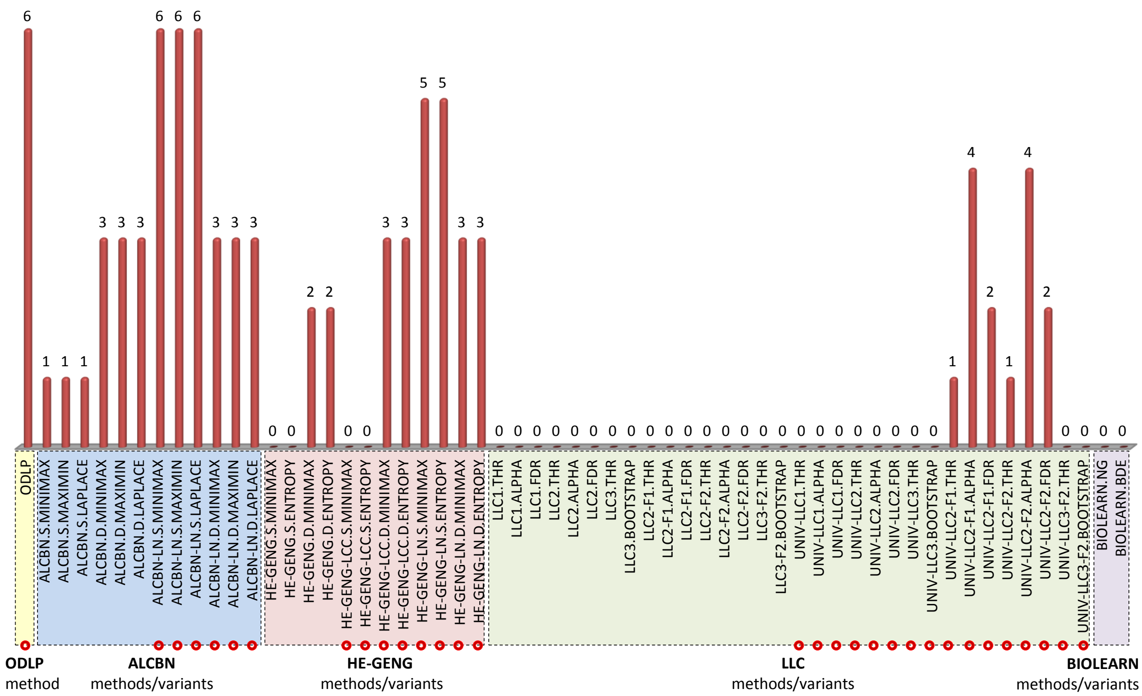


Figure 7: Number of local causal pathways discovered by an algorithm with optimal distance and with the same or fewer experiments than members of the pathway (out of 11 local causal pathways).Red circles denote methods designed for the discovery of local causal pathways. These include our modifications of the original global methods for local learning.

fewer pathways.

Table 3 uses data from Figures 5-8 for 58 methods/variants for active learning of causal networks to assess how the original global network learning methods (N = 28) perform relative to the methods modified specifically for local learning (N = 30). As can be seen, method variants modified for local learning fail in significantly fewer pathways, discover more pathways with optimal distance metric (reflecting structural discovery accuracy), and also achieve optimal distance metric with small number of experiments in more pathways than the original global learning methods/variants.

Analysis based on averages: The following analyses in Figures 9-11 visualize values of various metrics averaged over local causal pathway discovery tasks where all participating methods have completed and returned results (since we consider different number pathways from different networks, we first average results within each network and then over all networks). These analyses provide additional information compared to the counts of successes/failures because they also quantify the magnitude of successes/failures by reporting the average values. However, since only one method (ODLP) has completed on all 11 pathways, we have to a use a subset with 10 pathways (excluding P1M) and focus only

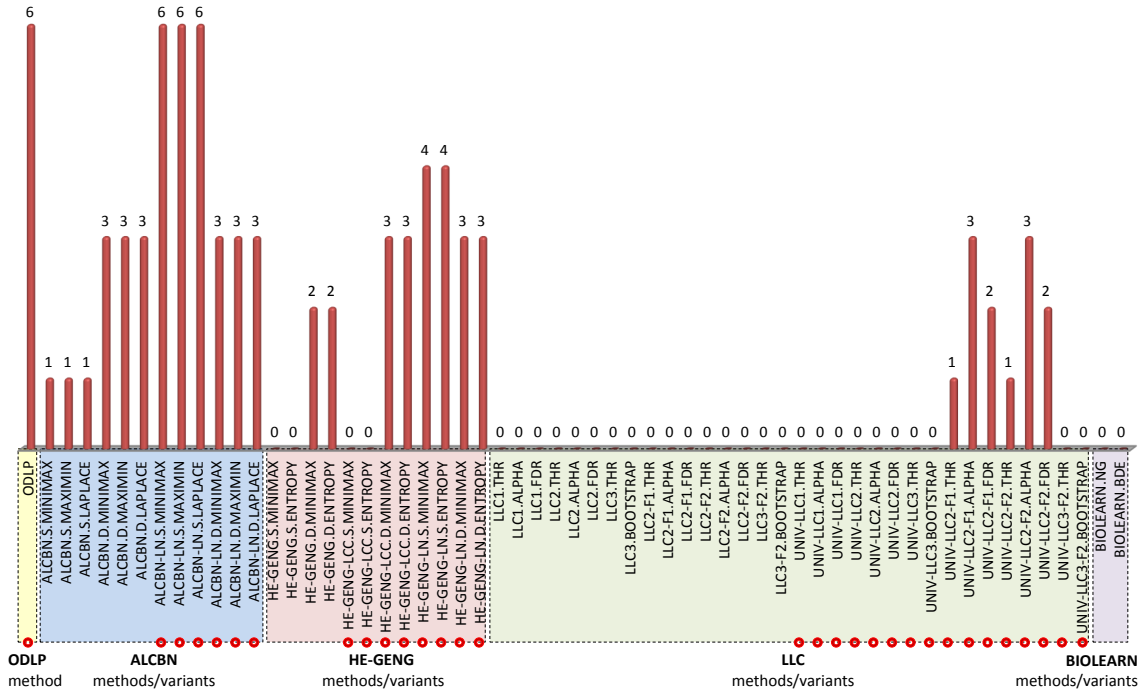


Figure 8: Number of local causal pathways discovered by an algorithm with optimal distance and with 10 or fewer experiments (out of 11 local causal pathways). Red circles denote methods designed for the discovery of local causal pathways. These include our modifications of the original global methods for local learning.

on 24 out of all 59 methods that have completed for all pathways in the considered subset.

Figure 9 shows a bull's eye plot for the distance metric and the number of experiments averaged over 10 local causal pathways. Location of the circles corresponds to values of the distance metric: the closer is circle to the center, the smaller (better) is the distance. The color of the circles corresponds to the number of experiments: the lighter is color, the more experiments are required. As can be seen, ODLP and a variant of LLC, UNIV-LLC3.THR, have the smallest average values of the distance metric, 9.6% and 12%, respectively. ODLP achieves this result with only 5 experiments, while the result of UNIV-LLC3.THR is based on 280 experiments. It is fair to note here that the ODLP method specifically optimizes the number of experiments, while UNIV-LLC3.THR uses experiments for all variables with significant univariate association with the target variable. An alternative and more detailed visualization of the data from Figure 9 is given in Figure 10 that shows a plot of distance versus number of experiments/number of variables in the network averaged over 10 local causal pathways.

Finally, Figure 11 shows a plot of sensitivity versus specificity averaged over 10 local causal pathways. A variant of LLC, UNIV-LLC3.THR, is the only method that has larger sensitivity than ODLP: sensitivity of ODLP and UNIV-LLC3.THR is 86.5% and 88.3%, respectively. However, this small 1.8% increase in sensitivity is accompanied by a significant

	Global Learning	Local Learning	P-value
Number of methods/variants	28	30	N.A.
Number of local causal pathways where the method was terminated/failed	Mean = 6.57 (St. dev. = 3.97)	Mean = 2.13 (St. dev. = 1.68)	6.33×10^{-7}
Number of local causal pathways discovered by a method with optimal distance (structural accuracy)	Mean = 1.61 (St. dev. = 1.73)	Mean = 3.10 (St. dev. = 1.56)	1.06×10^{-3}
Number of local causal pathways discovered by a method with optimal distance and with the same or fewer experiments than members of the pathway	Mean = 0.57 (St. dev. = 1.03)	Mean = 2.10 (St. dev. = 2.12)	1.09×10^{-3}
Number of local causal pathways discovered by a method with optimal distance and with 10 or fewer experiments	Mean = 0.57 (St. dev. = 1.03)	Mean = 1.97 (St. dev. = 1.99)	1.62×10^{-3}

Table 3: Comparison of performance of local and global learning methods/variants. P-values were obtained with a two-sample t-test. Statistical significance was assessed at 5% alpha level.

loss of specificity: specificity of ODLP is 99.97%, while specificity of UNIV-LLC3.THR is 90.4%. Finally, there are no methods that have larger specificity than ODLP.

6. Discussion

Methods for experimentally efficient and accurate discovery of local causal pathways from data can readily provide significant advances in many fields. For example, they can increase efficiency of drug discovery, facilitate development of socio-economic policies with desirable outcomes, or lead to successful marketing campaigns. Prior research has introduced several methods for active learning of the *entire/global* causal networks that utilize both observational and limited experimental data. The current study introduced new methods (termed ODLP) for discovery of local causal pathways around the target variable of interest using observational and experimental data, a topic not previously explored in the literature. Our new methods aim to minimize the number of experiments and also have substantially less restrictive theoretical assumptions for correctness compared to existing alternatives. An extensive empirical comparison of ODLP with 58 state-of-the-art methods/variants in high-dimensional datasets revealed that: (i) ODLP scales to datasets with 1,000,000 variables unlike comparator methods, which often fail to terminate within reasonable time even on datasets with of the order of 1,000 variables; (ii) ODLP achieves best local causal pathway discovery accuracy with minimal number of experiments compared to existing techniques

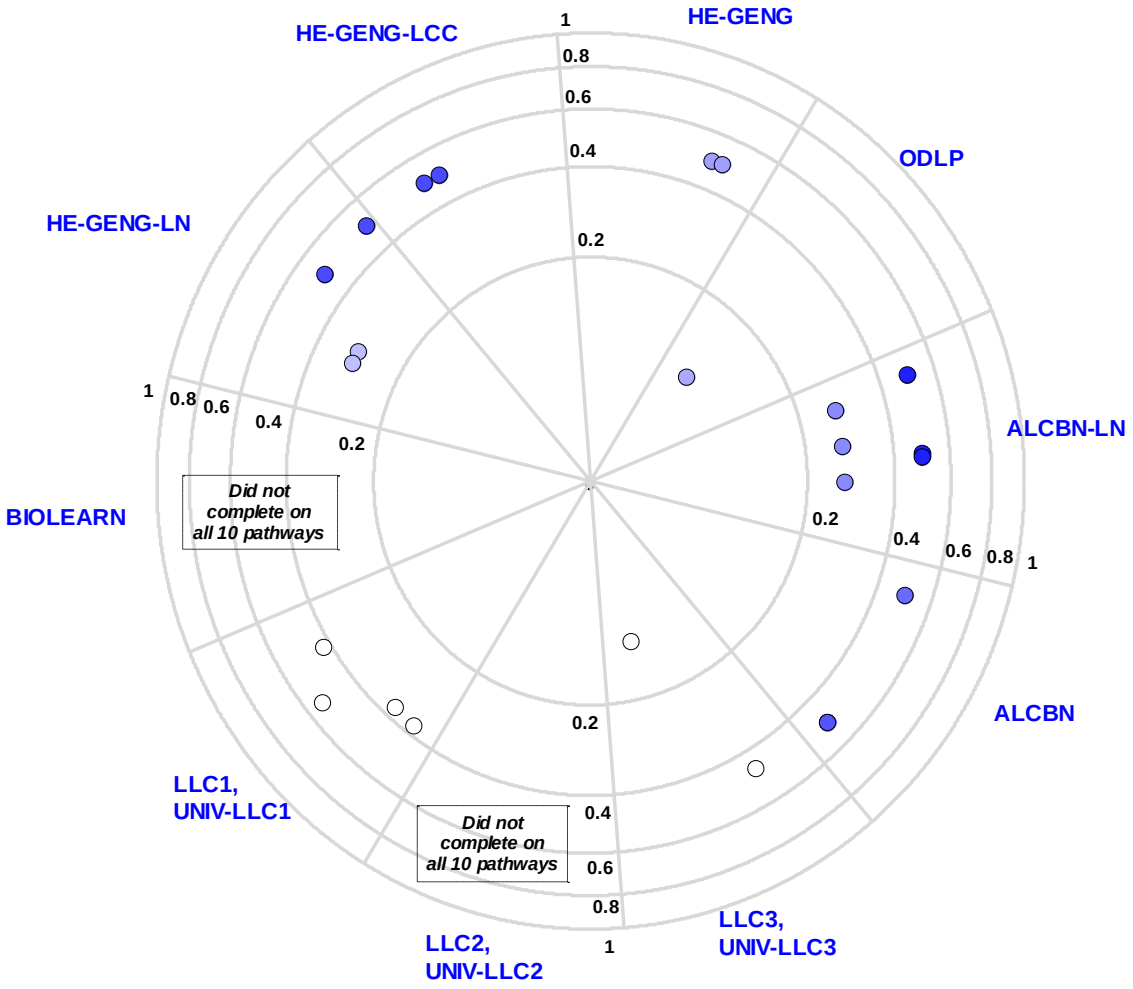


Figure 9: Bulls eye plot for the distance metric and the number of experiments averaged over 10 local causal pathways. Methods are denoted by circles. Location of the circles corresponds to values of the distance metric: the closer is circle to the center, the smaller (better) is the distance. Color of the circles corresponds to the number of experiments: the lighter is color, the more experiments are required.

under the assumption that all variables in the local neighborhood of the target are manipulable; and (iii) ODLP runs orders of magnitude faster than other methods (in most cases within 10-15 minutes for datasets with thousands of variables). A secondary contribution of this study is that we introduced local versions of prior methods for active learning of the entire/global causal networks so that the modified methods scale much better than the original techniques for this task.

There are several major directions for extending this work. First, further development of ODLP for situations when the target variable cannot be manipulated (e.g., it is a disease in humans) and therefore it is challenging to identify effects of the target variable. One

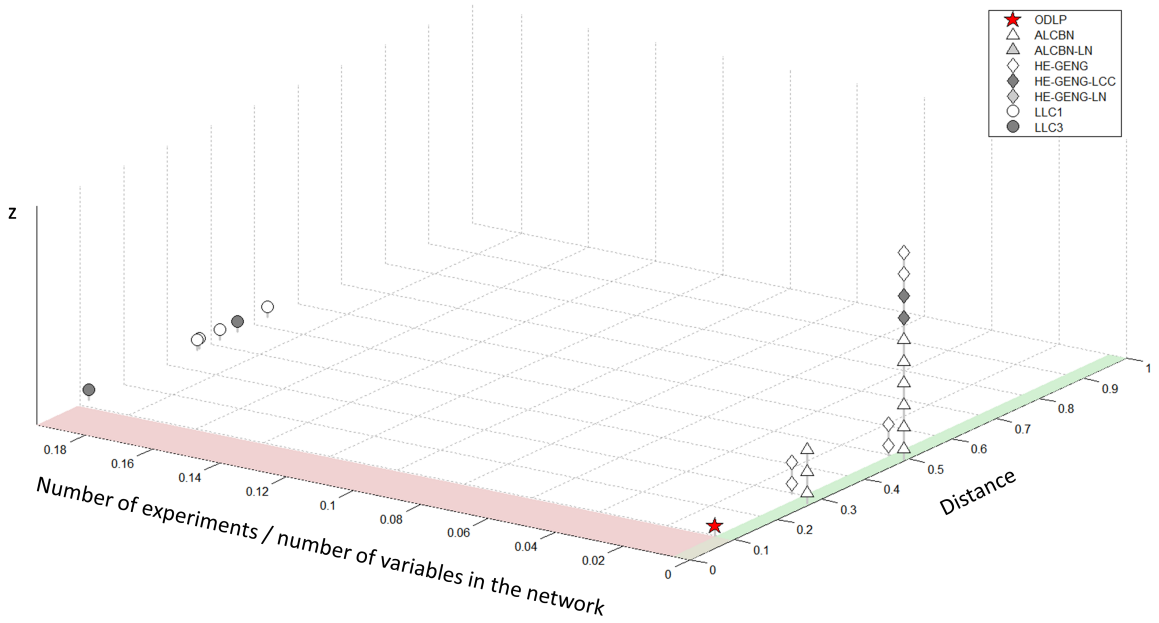


Figure 10: Distance versus number of experiments/number of variables in the network averaged over 10 local causal pathways. The vertical (z) dimension is used to produce a jitter plot so that multiple methods that have the same values of distance and number of experiments/number of variables in the network are not hidden in the graph. Methods located in the pale red area have smaller (better) distance than ODLP, and methods located in the pale green area require smaller number of experiments relative to the number of variables in the network.

possible strategy to solve this problem is to first identify all causes of the target variable and then identify effects through knowledge gained by manipulation of direct causes of the target variable. Second, extension of the ODLP method to work when there are hidden variables and/or feedback cycles. Related to this, the completeness of the algorithm can be improved by incorporating multi-variable manipulation experiments. Third, utilizing existing methods for causal orientation from non-experimental data to avoid unnecessary experiments, to the extent that these methods can produce accurate results in given distributions. These include both classical independence constraint-based (e.g., v-structure) techniques (Spirtes et al., 2000; Yin et al., 2008) or newer methods that can orient pairs of variables (Statnikov et al., 2012; Shimizu et al., 2006; Hoyer et al., 2009; Zhang and Hyvärinen, 2008; Daniusis et al., 2012; Janzing et al., 2012; Mooij et al., 2010). These newer methods could uncover the orientation of edges in non-linear (e.g. additive noise models (Hoyer et al., 2009)) or non-Gaussian (e.g. LinGAM (Shimizu et al., 2006)) cases, which are common in data from the biomedical domain. Fourth, further modifications of the existing state-of-the-art methods for active learning of the entire/global networks to adopt them for local causal pathway discovery task and seek to minimize the number of experiments. For instance, methods other than the PC algorithm could be implemented as

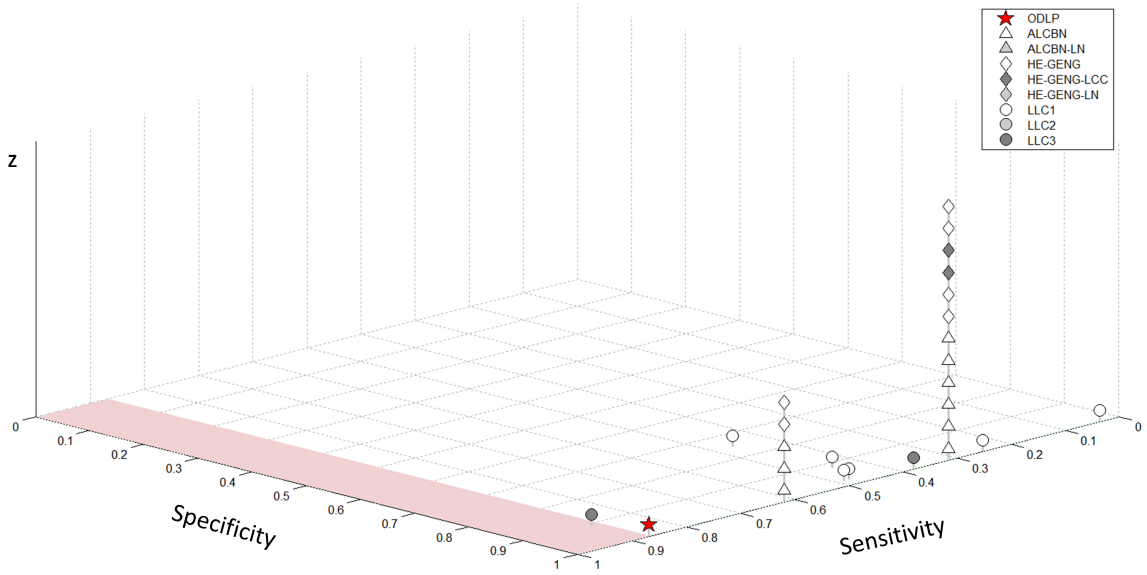


Figure 11: Sensitivity versus specificity averaged over 10 local causal pathways. The vertical (z) dimension is used to produce a jitter plot so that multiple methods that have the same values of sensitivity and specificity in the network are not hidden in the graph. Methods located in the pale red area have larger sensitivity than ODLP. There are no methods that have larger specificity than ODLP.

the starting point for edge orientation. The PC-Stable (Colombo and Maathuis, 2014) and GES algorithms (Chickering, 2003) might lead to increased accuracy, and Richardson's CCD Algorithm (Richardson, 1996) is applicable when acyclicity is not assumed. Finally fifth, extending the empirical comparison study to real (i.e., non-simulated) high-dimensional data. Use of real data is challenging because (i) for most large-scale systems the underlying causal relations are not known, and (ii) obtaining real experimental data is very expensive. Performing such studies in other domains (e.g., economics, marketing, ecology, etc.) is also worthwhile.

Acknowledgments

The evaluation of the methods in this work was supported in part by the grants 1UL1 RR029893 from the National Center for Research Resources and R01 LM011179-01A1 from the National Library of Medicine, National Institutes of Health. The authors thank Frederick Eberhardt and Antti Hyttinen for providing codes of the LLC methods and advice on running these algorithms in the empirical study.

Appendix A. List of Variants of the ALCBN, HE-GENG, and LLC Methods Used in This Work

ALCBN:
- First use the PC algorithm to learn an undirected or partially directed global graph from observational data (i.e., graph over all observed variables).
- Then orient edges by sequentially manipulating variables chosen by some decision criterion.

Method variant name	*Method variant description*
1. ***ALCBN.S.MINIMAX***	Starting from the undirected graph, use mini-max decision criterion to select variables for manipulation.
2. ***ALCBN.S.MAXIMIN***	Starting from the undirected graph, use maxi-min decision criterion to select variables for manipulation.
3. ***ALCBN.S.LAPLACE***	Starting from the undirected graph, use Laplace decision criterion to select variables for manipulation.
4. ***ALCBN.D.MINIMAX***	Starting from the partially directed graph, use mini-max decision criterion to select variables for manipulation.
5. ***ALCBN.D.MAXIMIN***	Starting from the partially directed graph, use maxi-min decision criterion to select variables for manipulation.
6. ***ALCBN.D.LAPLACE***	Starting from the partially directed graph, use Laplace decision criterion to select variables for manipulation.
7. ***ALCBN-LN.S.MINIMAX***	Same as *ALCBN.S.MINIMAX*, but select variables for manipulation only from the local causal pathway of the target.
8. ***ALCBN-LN.S.MAXIMIN***	Same as *ALCBN.S.MAXIMIN*, but select variables for manipulation only from the local causal pathway of the target.
9. ***ALCBN-LN.S.LAPLACE***	Same as *ALCBN.S.LAPLACE*, but select variables for manipulation only from the local causal pathway of the target.
10. ***ALCBN-LN.D.MINIMAX***	Same as *ALCBN.D.MINIMAX*, but select variables for manipulation only from the local causal pathway of the target.
11. ***ALCBN-LN.D.MAXIMIN***	Same as *ALCBN.D.MAXIMIN*, but select variables for manipulation only from the local causal pathway of the target.
12. ***ALCBN-LN.D.LAPLACE***	Same as *ALCBN.D.LAPLACE*, but select variables for manipulation only from the local causal pathway of the target.

Method of He and Geng (HE-GENG):
- First use the PC algorithm to learn an undirected or partially directed global graph from observational data (i.e., graph over all observed variables).
- Then orient edges in each obtained chain component separately by sequentially manipulating variables chosen by some decision criterion.

Method variant name	*Method variant description*
1. ***HE-GENG.S.MINIMAX***	Starting from the undirected graph, use mini-max decision criterion to select variables for manipulation.
2. ***HE-GENG.S.ENTROPY***	Starting from the undirected graph, use maxi-min entropy decision criterion to select variables for manipulation.
3. ***HE-GENG.D.MINIMAX***	Starting from the partially directed graph, use mini-max decision criterion to select variables for manipulation.
4. ***HE-GENG.D.ENTROPY***	Starting from the partially directed graph, use maximum entropy decision criterion to select variables for manipulation.
5. ***HE-GENG-LCC.S.MINIMAX***	Same as *HE-GENG.S.MINIMAX*, but select variables for manipulation only from the local chain component of the target.
6. ***HE-GENG-LCC.S.ENTROPY***	Same as *HE-GENG.S.ENTROPY*, but select variables for manipulation only from the local chain component of the target.
7. ***HE-GENG-LCC.D.MINIMAX***	Same as *HE-GENG.D.MINIMAX*, but select variables for manipulation only from the local chain component of the target.
8. ***HE-GENG-LCC.D.ENTROPY***	Same as *HE-GENG.D.ENTROPY*, but select variables for manipulation only from the local chain component of the target.
9. ***HE-GENG-LN.S.MINIMAX***	Same as *HE-GENG.S.MINIMAX*, but select variables for manipulation only from the local causal pathway of the target.
10. ***HE-GENG-LN.S.ENTROPY***	Same as *HE-GENG.S.ENTROPY*, but select variables for manipulation only from the local causal pathway of the target.
11. ***HE-GENG-LN.D.MINIMAX***	Same as *HE-GENG.D.MINIMAX*, but select variables for manipulation only from the local causal pathway of the target.
12. ***HE-GENG-LN.D.ENTROPY***	Same as *HE-GENG.D.ENTROPY*, but select variables for manipulation only from the local causal pathway of the target.

Table A1: Conditional independence constraint-based structure learning methods/variants used in this work. Modifications of the original methods that focus on discovery of local causality are highlighted.

LLC:

- Assume linear relations between variables. These relations can be represented by an "effects matrix" where each element is the coefficient of the linear relation between variables.
- From each manipulated dataset, derive constraints on the effects matrix which are combined into a linear system (we refer to these constraints as "main constraints").
- Additionally assuming faithfulness allows:
  - utilizing PC algorithm on manipulated data and possibly observational data to learn adjacencies between variables. Non-adjacent variables imply additional constraints on the effects matrix that are added to the linear system (we refer to these constraints as "0-constraints").
  - defining an optimal order of variables for manipulation geared towards identification of the effects matrix.
- Solve the above linear system to identify the effects matrix.
- Elements in the effects matrix correspond to coefficients of the underlying linear relations.
- Filter the effects matrix to obtain edges in the output graph using one of the following methods:
  - THR: Obtain edges by applying a threshold of 0.1 on the coefficients of the identified effects matrix.
  - ALPHA: Using 100 data permutations, estimate the null distribution of the coefficients of the effects matrix. Obtain edges by choosing significant coefficients at 5% alpha level.
  - FDR: Using 100 data permutations, estimate the null distribution of the coefficients of the effects matrix. Obtain edges by choosing significant coefficients at 5% FDR level.
  - BOOTSTRAP: Identify effects matrix in 30 datasets sampled from the original data with replacement. Obtain edges by choosing elements of the effects matrix whose mean coefficient over resampled datasets is higher than the standard deviation.

Method variant name	*Method variant description*
1. ***LLC1.THR***	Manipulate all variables associated with the target to obtain manipulated data. Derive main constraints on the effects matrix and solve the linear system using the method LLC1. Find edges in graph by method THR.
2. ***LLC1.ALPHA***	Same as *LLC1.THR*, but use method ALPHA to find edges in graph.
3. ***LLC1.FDR***	Same as *LLC1.THR*, but use method FDR to find edges in graph.
4. ***LLC2.THR***	Same as *LLC1.THR*, but use LLC2 method to derive main constraints on the effects matrix and solve the linear system.
5. ***LLC2.ALPHA***	Same as *LLC1.THR*, but use LLC2 method to derive main constraints on the effects matrix and solve the linear system and method ALPHA to find edges in graph.
6. ***LLC2.FDR***	Same as *LLC1.THR*, but use LLC2 method to derive main constraints on the effects matrix and solve the linear system and method FDR to find edges in graph.
7. ***LLC3.THR***	Same as *LLC1.THR*, but use LLC3 method to derive main constraints on the effects matrix and solve the linear system.
8. ***LLC3.BOOTSTRAP***	Same as *LLC1.THR*, but use LLC3 method to derive main constraints on the effects matrix and solve the linear system and method BOOTSTRAP to find edges in graph.
9. ***LLC2-F1.THR***	Manipulate a random variable to obtain manipulated data. Apply PC algorithm on manipulated data to obtain 0-constraints on the effects matrix. Derive main constraints on the effects matrix and solve the linear system using the method LLC2. Determine optimal variable for manipulation. Repeat the above steps until the effects matrix has been identified. Find edges in graph by method THR.
10. ***LLC2-F1.ALPHA***	Same as *LLC2-F1.THR*, but use method ALPHA to find edges in graph.
11. ***LLC2-F1.FDR***	Same as *LLC2-F1.THR*, but use method FDR to find edges in graph.
12. ***LLC2-F2.THR***	Same as *LLC2-F1.THR*, but apply PC to both observational and manipulated data to obtain 0-constraints on the effects matrix.
13. ***LLC2-F2.ALPHA***	Same as *LLC2-F1.THR*, but apply PC to both observational and manipulated data to obtain 0-constraints on the effects matrix and method ALPHA to find edges in graph.
14. ***LLC2-F2.FDR***	Same as *LLC2-F1.THR*, but apply PC to both observational and manipulated data to obtain 0-constraints on the effects matrix and method FDR to find edges in graph.
15. ***LLC3-F2.THR***	Same as *LLC2-F1.THR*, but apply PC to both observational and manipulated data to obtain 0-constraints on the effects matrix and method LLC3 to derive main constraints on the effects matrix and solve the linear system.
16. ***LLC3-F2.BOOTSTRAP***	Same as *LLC2-F1.THR*, but apply PC to both observational and manipulated data to obtain 0-constraints on the effects matrix, method LLC3 to derive main constraints on the effects matrix and solve the linear system, and method BOOTSTRAP to find edges in graph.
17. ***UNIV-LLC1.THR***	18. ***UNIV-LLC2.ALPHA***	19. ***UNIV-LLC2-F1.THR***	20. ***UNIV-LLC2-F2.ALPHA***	Same as above methods without prefix "UNIV", except for using only variables that are univariately associated with the target to identify the effects matrix. All other variables are not considered at all by the method.
21. ***UNIV-LLC1.ALPHA***	22. ***UNIV-LLC2.FDR***	23. ***UNIV-LLC2-F1.ALPHA***	24. ***UNIV-LLC2-F2.FDR***	
25. ***UNIV-LLC1.FDR***	26. ***UNIV-LLC3.THR***	27. ***UNIV-LLC2-F1.FDR***	28. ***UNIV-LLC3-F2.THR***	
29. ***UNIV-LLC2.THR***	30. ***UNIV-LLC3.BOOTSTRAP***	31. ***UNIV-LLC2-F2.THR***	32. ***UNIV-LLC3-F2.BOOTSTRAP***	

Table A2: Linear cyclic models-based structure learning methods/variants used in this work. Modifications of the original methods that focus on discovery of local causality are highlighted.

Appendix B. Information about P1000 and P1M Networks

Vertex	Parent	Coefficient	Noise Coefficient
1	2	0.9	0
2	40	0.8	0.2
	41	0.8	0.2
3	6	0.6	0
4	6	0.8	0
5	2	0.8	0
6	2	0.9	0
7	8	0.9	0
8	9	1.1	0
9	39	0.9	0
10	9	0.8	0
11	9	0.7	0
13	12	0.6	0
14	13	0.8	0
15	16	0.7	0
16	12	0.9	0
17	15	0.9	0
18	54	0.7	0.2
19	18	0.9	0
20	19	0.1	0
21	54	0.6	0.1
22	21	0.9	0
23	22	0.2	0
24	23	0.4	0.3
26	25	0.9	0.2
	17	0.6	0.2

Vertex	Parent	Coefficient	Noise Coefficient
27	26	0.4	0.1
28	17	0.5	0.1
29	30	0.1	0.1
30	31	0.7	0.2
31	32	0.9	0.1
32	35	0.6	0.1
33	35	0.9	0.2
34	35	0.5	0.3
35	36	0.1	0.2
	37	0.6	0.2
37	38	0.8	0.2
38	39	0.1	0.1
40	39	0.5	0.2
47	44	0.7	0.3
48	45	0.9	0.3
49	46	0.1	0.1
50	48	0.3	0.2
	49	0.4	0.2
51	20	0.9	0.1
	50	0.4	0.1
52	20	0.6	0.2
	53	0.8	0.2
54 (T)	1	0.3	0.1
	7	0.3	0.1
	12	0.3	0.1

Table B1: Parameterization of the P1000 network. Data for a given vertex/variable V is a linear combination of its parents and Gaussian noise: $V = \sum_p (Coef_{parent_p} + N(0, Coef_{noise_p}))$. The data for vertices without any parents was sampled from Gaussian distribution $N(0, 1)$ and is not shown in the following table.

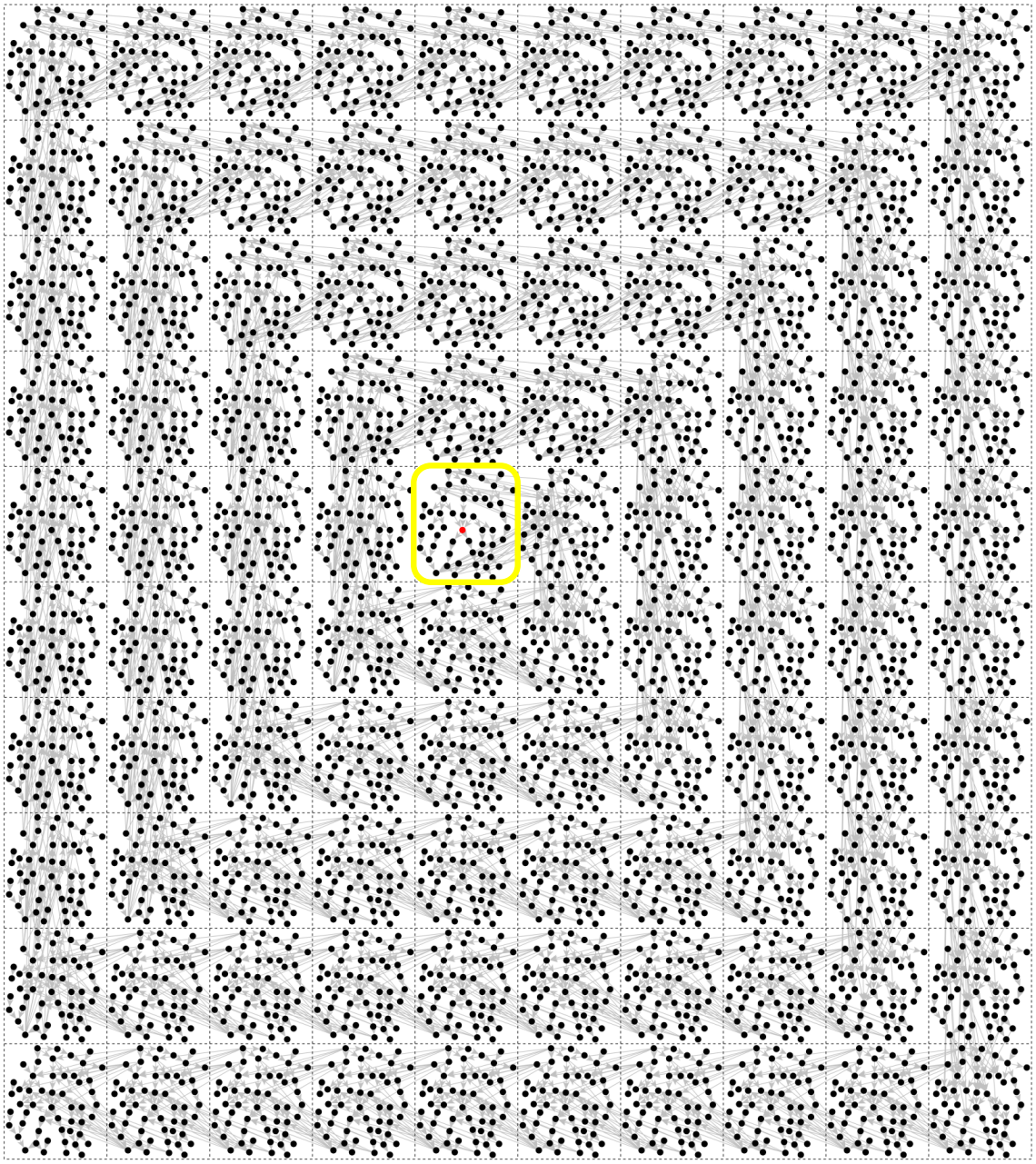


Figure B1: A fragment of the P1M network. The little red dot in the middle (in the tile with yellow outline) represents the target variable, black dots represent other variables. Only the connected components of the first 100 tiles were shown.

Appendix C. Detailed Results of Empirical Experiments

	REGED				P1000				P1M			
Method Name	Sensitivity	Specificity	Distance	N exp	Sensitivity	Specificity	Distance	N exp	Sensitivity	Specificity	Distance	N exp
ODLP	86.7%	100.0%	9.4%	1	100.0%	100.0%	0.0%	18	100.0%	100.0%	0.0%	19
ALCBN.S.MINIMAX	86.7%	100.0%	9.4%	47	0.0%	99.8%	70.7%	577	T1			
ALCBN.S.MAXIMIN	86.7%	100.0%	9.4%	91	0.0%	99.8%	70.7%	368	T1			
ALCBN.S.LAPLACE	86.7%	100.0%	9.4%	62	0.0%	99.8%	70.7%	442	T1			
ALCBN.D.MINIMAX	26.7%	99.6%	51.9%	0	0.0%	99.8%	70.7%	0	T1			
ALCBN.D.MAXIMIN	26.7%	99.6%	51.9%	0	0.0%	99.8%	70.7%	0	T1			
ALCBN.D.LAPLACE	26.7%	99.6%	51.9%	0	0.0%	99.8%	70.7%	0	T1			
ALCBN-LN.S.MINIMAX	86.7%	100.0%	9.4%	1	0.0%	99.8%	70.7%	1	T1			
ALCBN-LN.S.MAXIMIN	86.7%	100.0%	9.4%	1	0.0%	99.8%	70.7%	1	T1			
ALCBN-LN.S.LAPLACE	86.7%	100.0%	9.4%	1	0.0%	99.8%	70.7%	1	T1			
ALCBN-LN.D.MINIMAX	26.7%	99.6%	51.9%	0	0.0%	99.8%	70.7%	0	T1			
ALCBN-LN.D.MAXIMIN	26.7%	99.6%	51.9%	0	0.0%	99.8%	70.7%	0	T1			
ALCBN-LN.D.LAPLACE	26.7%	99.6%	51.9%	0	0.0%	99.8%	70.7%	0	T1			
HE-GENG.S.MINIMAX	86.7%	100.0%	9.4%	337	0.0%	99.8%	70.7%	32	T1			
HE-GENG.S.ENTROPY	86.7%	100.0%	9.4%	337	0.0%	99.8%	70.7%	32	T1			
HE-GENG.D.MINIMAX	26.7%	99.6%	51.9%	0	0.0%	99.8%	70.7%	0	T1			
HE-GENG.D.ENTROPY	26.7%	99.6%	51.9%	0	0.0%	99.8%	70.7%	0	T1			
HE-GENG-LCC.S.MINIMAX	86.7%	100.0%	9.4%	108	0.0%	99.8%	70.7%	63	T1			
HE-GENG-LCC.S.ENTROPY	86.7%	100.0%	9.4%	108	0.0%	99.8%	70.7%	63	T1			
HE-GENG-LCC.D.MINIMAX	26.7%	99.6%	51.9%	0	0.0%	99.8%	70.7%	0	T1			
HE-GENG-LCC.D.ENTROPY	26.7%	99.6%	51.9%	0	0.0%	99.8%	70.7%	0	T1			
HE-GENG-LN.S.MINIMAX	86.7%	100.0%	9.4%	13	0.0%	99.8%	70.7%	5	T1			
HE-GENG-LN.S.ENTROPY	86.7%	100.0%	9.4%	13	0.0%	99.8%	70.7%	5	T1			
HE-GENG-LN.D.MINIMAX	26.7%	99.6%	51.9%	0	0.0%	99.8%	70.7%	0	T1			
HE-GENG-LN.D.ENTROPY	26.7%	99.6%	51.9%	0	0.0%	99.8%	70.7%	0	T1			
LLC1.THR	86.7%	50.5%	36.3%	540	100.0%	51.1%	34.6%	85	T4			
LLC1.ALPHA	6.7%	98.8%	66.0%	540	0.0%	99.9%	70.7%	85	T4			
LLC1.FDR	6.7%	99.7%	66.0%	540	0.0%	100.0%	70.7%	85	T4			
LLC2.THR	T1				0.0%	100.0%	70.7%	85	T4			
LLC2.ALPHA	T2				T2				T4			
LLC2.FDR	T2				T2				T4			
LLC3.THR	0.0%	100.0%	70.7%	540	0.0%	100.0%	70.7%	85	T4			
LLC3.BOOTSTRAP	100.0%	93.8%	4.4%	540	100.0%	69.9%	21.3%	85	T4			
LLC2-F1.THR	T1				T1				T1			
LLC2-F1.ALPHA	T2				T2				T2			
LLC2-F1.FDR	T2				T2				T2			
LLC2-F2.THR	T1				T1				T1			
LLC2-F2.ALPHA	T2				T2				T2			
LLC2-F2.FDR	T2				T2				T2			
LLC3-F2.THR	T4				T4				T4			
LLC3-F2.BOOTSTRAP	T4				T4				T4			
UNIV-LLC1.THR	80.0%	73.6%	23.4%	540	100.0%	95.0%	3.5%	85	T4			
UNIV-LLC1.ALPHA	6.7%	98.8%	66.0%	540	0.0%	99.5%	70.7%	85	T4			
UNIV-LLC1.FDR	6.7%	99.7%	66.0%	540	0.0%	100.0%	70.7%	85	T4			
UNIV-LLC2.THR	0.0%	100.0%	70.7%	540	100.0%	98.5%	1.1%	85	T4			
UNIV-LLC2.ALPHA	T2				60.0%	99.9%	28.3%	85	T4			
UNIV-LLC2.FDR	T2				40.0%	100.0%	42.4%	85	T4			
UNIV-LLC3.THR	80.0%	73.8%	23.3%	540	100.00%	93.48%	4.6%	85	T4			
UNIV-LLC3.BOOTSTRAP	13.3%	100.0%	61.3%	540	0.00%	100.00%	70.7%	85	T4			
UNIV-LLC2-F1.THR	0.0%	100.0%	70.7%	4	0.0%	100.0%	70.7%	5	T1			
UNIV-LLC2-F1.ALPHA	T2				40.0%	99.1%	42.4%	5	T2			
UNIV-LLC2-F1.FDR	T2				0.0%	100.0%	70.7%	5	T2			
UNIV-LLC2-F2.THR	T1				0.0%	99.9%	70.7%	2	T1			
UNIV-LLC2-F2.ALPHA	T2				40.0%	97.2%	42.5%	2	T2			
UNIV-LLC2-F2.FDR	T2				20.0%	99.7%	56.6%	2	T2			
UNIV-LLC3-F2.THR	T4				0.0%	100.0%	70.7%	11	T4			
UNIV-LLC3-F2.BOOTSTRAP	T4				100.0%	96.8%	2.2%	11	T4			
BIOLEARN.NG	T3				0.0%	99.6%	70.7%	85	T3			
BIOLEARN.BDE	T3				20.0%	100.0%	56.6%	85	T3			

Explanation of termination/ failure codes:

T1 = Experiments when the algorithm was terminated after 30 days of single-core time limit for tasks that cannot be easily parallelized;

T2 = Experiments when the algorithm was terminated after 3,000 day multi-core time limit (spread over 100 cores) for tasks that can be easily parallelized;

T3 = Experiments when the authors' implementation of the algorithm failed for unknown reason;

T4 = Experiments when the algorithm required more than 48 GB RAM.

Table C1: Detailed results of experiments for REGED, P1000, and P1M networks.

	ECOLI (agaR)				ECOLI (allR)				ECOLI (zur)				ECOLI (lexA)			
Method Name	Sensitivity	Specificity	Distance	N exp	Sensitivity	Specificity	Distance	N exp	Sensitivity	Specificity	Distance	N exp	Sensitivity	Specificity	Distance	N exp
ODLP	87.5%	99.9%	8.8%	1	100.0%	99.9%	0.1%	1	100.0%	99.9%	0.1%	3	90.7%	99.9%	6.6%	1
ALCBN.S.MINIMAX	87.5%	100.0%	8.8%	264	100.0%	99.9%	0.1%	162	100.0%	99.9%	0.1%	213	90.7%	99.9%	6.6%	4
ALCBN.S.MAXIMIN	87.5%	100.0%	8.8%	269	100.0%	99.9%	0.1%	436	100.0%	99.9%	0.1%	288	90.7%	99.9%	6.6%	4
ALCBN.S.LAPLACE	87.5%	100.0%	8.8%	212	100.0%	99.9%	0.1%	143	100.0%	99.9%	0.1%	292	90.7%	99.9%	6.6%	4
ALCBN.D.MINIMAX	100.0%	100.0%	0.0%	0	100.0%	99.9%	0.1%	0	100.0%	99.9%	0.1%	4	3.7%	98.3%	68.1%	0
ALCBN.D.MAXIMIN	100.0%	100.0%	0.0%	0	100.0%	99.9%	0.1%	0	100.0%	99.9%	0.1%	4	3.7%	98.3%	68.1%	0
ALCBN.D.LAPLACE	100.0%	100.0%	0.0%	0	100.0%	99.9%	0.1%	0	100.0%	99.9%	0.1%	3	3.7%	98.3%	68.1%	0
ALCBN-LN.S.MINIMAX	87.5%	100.0%	8.8%	1	100.0%	99.9%	0.1%	1	100.0%	99.9%	0.1%	1	90.7%	99.9%	6.6%	1
ALCBN-LN.S.MAXIMIN	87.5%	100.0%	8.8%	1	100.0%	99.9%	0.1%	1	100.0%	99.9%	0.1%	1	90.7%	99.9%	6.6%	1
ALCBN-LN.S.LAPLACE	87.5%	100.0%	8.8%	1	100.0%	99.9%	0.1%	1	100.0%	99.9%	0.1%	1	90.7%	99.9%	6.6%	1
ALCBN-LN.D.MINIMAX	100.0%	100.0%	0.0%	0	100.0%	99.9%	0.1%	0	100.0%	99.9%	0.1%	1	3.7%	98.3%	68.1%	0
ALCBN-LN.D.MAXIMIN	100.0%	100.0%	0.0%	0	100.0%	99.9%	0.1%	0	100.0%	99.9%	0.1%	1	3.7%	98.3%	68.1%	0
ALCBN-LN.D.LAPLACE	100.0%	100.0%	0.0%	0	100.0%	99.9%	0.1%	0	100.0%	99.9%	0.1%	1	3.7%	98.3%	68.1%	0
HE-GENG.S.MINIMAX	87.5%	100.0%	8.8%	86	100.0%	99.9%	0.1%	25	T1				T1			
HE-GENG.S.ENTROPY	87.5%	100.0%	8.8%	86	100.0%	99.9%	0.1%	25	T1				T1			
HE-GENG.D.MINIMAX	100.0%	100.0%	0.0%	0	100.0%	99.9%	0.1%	0	100.0%	99.9%	0.1%	74	3.7%	98.3%	68.1%	0
HE-GENG.D.ENTROPY	100.0%	100.0%	0.0%	0	100.0%	99.9%	0.1%	0	100.0%	99.9%	0.1%	74	3.7%	98.3%	68.1%	0
HE-GENG-LCC.S.MINIMAX	87.5%	100.0%	8.8%	86	100.0%	99.9%	0.1%	25	T1				T1			
HE-GENG-LCC.S.ENTROPY	87.5%	100.0%	8.8%	86	100.0%	99.9%	0.1%	25	T1				T1			
HE-GENG-LCC.D.MINIMAX	100.0%	100.0%	0.0%	0	100.0%	99.9%	0.1%	0	100.0%	99.9%	0.1%	5	3.7%	98.3%	68.1%	0
HE-GENG-LCC.D.ENTROPY	100.0%	100.0%	0.0%	0	100.0%	99.9%	0.1%	0	100.0%	99.9%	0.1%	5	3.7%	98.3%	68.1%	0
HE-GENG-LN.S.MINIMAX	87.5%	100.0%	8.8%	2	100.0%	99.9%	0.1%	6	100.0%	99.9%	0.1%	2	77.8%	99.6%	15.7%	30
HE-GENG-LN.S.ENTROPY	87.5%	100.0%	8.8%	2	100.0%	99.9%	0.1%	6	100.0%	99.9%	0.1%	2	77.8%	99.6%	15.7%	30
HE-GENG-LN.D.MINIMAX	100.0%	100.0%	0.0%	0	100.0%	99.9%	0.1%	0	100.0%	99.9%	0.1%	5	3.7%	98.3%	68.1%	0
HE-GENG-LN.D.ENTROPY	100.0%	100.0%	0.0%	0	100.0%	99.9%	0.1%	0	100.0%	99.9%	0.1%	5	3.7%	98.3%	68.1%	0
LLC1.THR	0.0%	100.0%	70.7%	82	0.0%	100.0%	70.7%	88	0.0%	100.0%	70.7%	90	0.0%	100.0%	70.7%	147
LLC1.ALPHA	100.0%	49.8%	35.5%	82	100.0%	50.3%	35.2%	88	100.0%	50.6%	35.0%	90	96.3%	51.7%	34.2%	147
LLC1.FDR	100.0%	51.3%	34.4%	82	100.0%	51.4%	34.3%	88	100.0%	51.6%	34.2%	90	96.3%	52.8%	33.5%	147
LLC2.THR	T1				T1				T1				T1			
LLC2.ALPHA	T2				T2				T2				T2			
LLC2.FDR	T2				T2				T2				T2			
LLC3.THR	0.0%	100.0%	70.7%	82	10.0%	100.0%	63.6%	88	0.0%	100.0%	70.7%	90	0.0%	100.0%	70.7%	147
LLC3.BOOTSTRAP	100.0%	67.2%	23.2%	82	100.0%	72.0%	19.8%	88	100.0%	69.4%	21.7%	90	98.2%	78.6%	15.2%	147
LLC2-F1.THR	T1				T1				T1				T1			
LLC2-F1.ALPHA	T2				T2				T2				T2			
LLC2-F1.FDR	T2				T2				T2				T2			
LLC2-F2.THR	T1				T1				T1				T1			
LLC2-F2.ALPHA	T2				T2				T2				T2			
LLC2-F2.FDR	T2				T2				T2				T2			
LLC3-F2.THR	T4				T4				T4				T4			
LLC3-F2.BOOTSTRAP	T4				T4				T4				T4			
UNIV-LLC1.THR	0.0%	100.0%	70.7%	82	0.0%	100.0%	70.7%	88	0.0%	100.0%	70.7%	90	0.0%	100.0%	70.7%	147
UNIV-LLC1.ALPHA	100.0%	99.7%	0.3%	82	100.0%	99.7%	0.3%	88	100.0%	99.3%	0.5%	90	94.4%	99.5%	3.9%	147
UNIV-LLC1.FDR	87.5%	100.0%	8.8%	82	100.0%	99.9%	0.1%	88	100.0%	100.0%	0.0%	90	90.7%	100.0%	6.6%	147
UNIV-LLC2.THR	0.0%	100.0%	70.7%	82	0.0%	100.0%	70.7%	88	0.0%	100.0%	70.7%	90	0.0%	100.0%	70.7%	147
UNIV-LLC2.ALPHA	100.0%	99.7%	0.3%	82	100.0%	99.7%	0.3%	88	100.0%	99.3%	0.5%	90	94.4%	99.5%	3.9%	147
UNIV-LLC2.FDR	87.5%	100.0%	8.8%	82	100.0%	99.9%	0.1%	88	100.0%	100.0%	0.0%	90	90.7%	100.0%	6.6%	147
UNIV-LLC3.THR	100.0%	98.3%	1.2%	82	100.0%	98.6%	1.0%	88	100.0%	98.3%	1.2%	90	87.0%	97.7%	9.3%	147
UNIV-LLC3.BOOTSTRAP	100.0%	98.0%	1.4%	82	100.0%	98.6%	1.0%	88	100.0%	97.8%	1.5%	90	88.9%	98.3%	8.0%	147
UNIV-LLC2-F1.THR	0.0%	100.0%	70.7%	5	0.0%	100.0%	70.7%	7	50.0%	99.9%	35.4%	6	0.0%	100.0%	70.7%	12
UNIV-LLC2-F1.ALPHA	75.0%	99.7%	17.7%	5	100.0%	99.1%	0.6%	7	100.0%	99.7%	0.2%	6	96.3%	96.4%	3.6%	12
UNIV-LLC2-F1.FDR	62.5%	99.9%	26.5%	5	100.0%	99.8%	0.1%	7	100.0%	99.8%	0.1%	6	72.2%	99.5%	19.7%	12
UNIV-LLC2-F2.THR	37.5%	99.9%	44.2%	4	0.0%	100.0%	70.7%	6	83.3%	99.8%	11.8%	5	0.0%	100.0%	70.7%	11
UNIV-LLC2-F2.ALPHA	50.0%	99.8%	35.4%	4	100.0%	99.0%	0.7%	6	100.0%	99.6%	0.3%	5	98.2%	96.3%	2.9%	11
UNIV-LLC2-F2.FDR	50.0%	99.9%	35.4%	4	100.0%	99.9%	0.1%	6	83.3%	99.8%	11.8%	5	66.7%	99.6%	23.6%	11
UNIV-LLC3-F2.THR	0.0%	100.0%	70.7%	17	0.0%	100.0%	70.7%	27	0.0%	100.0%	70.7%	32	0.0%	100.0%	70.7%	57
UNIV-LLC3-F2.BOOTSTRAP	75.0%	98.6%	17.7%	19	70.0%	99.1%	21.2%	25	83.3%	98.1%	11.9%	30	59.3%	97.5%	28.9%	46
BIOLEARN.NG	75.0%	99.9%	17.7%	82	50.0%	99.9%	35.4%	88	66.7%	99.9%	23.6%	90	90.7%	99.9%	6.6%	147
BIOLEARN.BDE	0.0%	99.8%	70.7%	82	0.0%	99.9%	70.7%	88	16.7%	99.9%	58.9%	90	50.0%	100.0%	35.4%	147

Table C2: Detailed results of experiments for ECOLI network (4 local causal neighborhoods). See Table C1 for explanation of termination/failure codes T1, T2, T3, and T4.

	YEAST (YBL005W)				YEAST (YFL044C)				YEAST (YLR014C)				YEAST (YKL112W)			
Method Name	Sensitivity	Specificity	Distance	N exp	Sensitivity	Specificity	Distance	N exp	Sensitivity	Specificity	Distance	N exp	Sensitivity	Specificity	Distance	N exp
ODLP	66.7%	99.93%	23.6%	1	66.7%	100.0%	23.6%	1	64.5%	100.0%	25.1%	1	61.0%	99.9%	27.6%	1
ALCBN.S.MINIMAX	T1				T1				T1				61.0%	99.9%	27.6%	1
ALCBN.S.MAXIMIN	T1				T1				T1				61.0%	99.9%	27.6%	1
ALCBN.S.LAPLACE	T1				T1				T1				61.0%	99.9%	27.6%	1
ALCBN.D.MINIMAX	6.7%	99.7%	66.0%	0	26.7%	99.9%	51.9%	0	35.5%	99.9%	45.6%	21	5.3%	98.0%	67.0%	0
ALCBN.D.MAXIMIN	6.7%	99.7%	66.0%	0	26.7%	99.9%	51.9%	0	35.5%	99.9%	45.6%	24	5.3%	98.0%	67.0%	0
ALCBN.D.LAPLACE	6.7%	99.7%	66.0%	0	26.7%	99.9%	51.9%	0	35.5%	99.9%	45.6%	20	5.3%	98.0%	67.0%	0
ALCBN-LN.S.MINIMAX	66.7%	99.9%	23.6%	1	66.7%	100.0%	23.6%	1	64.5%	100.0%	25.1%	1	61.0%	99.9%	27.6%	1
ALCBN-LN.S.MAXIMIN	66.7%	99.9%	23.6%	1	66.7%	100.0%	23.6%	1	64.5%	100.0%	25.1%	1	61.0%	99.9%	27.6%	1
ALCBN-LN.S.LAPLACE	66.7%	99.9%	23.6%	1	66.7%	100.0%	23.6%	1	64.5%	100.0%	25.1%	1	61.0%	99.9%	27.6%	1
ALCBN-LN.D.MINIMAX	6.7%	99.7%	66.0%	0	26.7%	99.9%	51.9%	0	35.5%	99.9%	45.6%	1	5.3%	98.0%	67.0%	0
ALCBN-LN.D.MAXIMIN	6.7%	99.7%	66.0%	0	26.7%	99.9%	51.9%	0	35.5%	99.9%	45.6%	1	5.3%	98.0%	67.0%	0
ALCBN-LN.D.LAPLACE	6.7%	99.7%	66.0%	0	26.7%	99.9%	51.9%	0	35.5%	99.9%	45.6%	1	5.3%	98.0%	67.0%	0
HE-GENG.S.MINIMAX	T1				T1				T1				T1			
HE-GENG.S.ENTROPY	T1				T1				T1				T1			
HE-GENG.D.MINIMAX	6.7%	99.7%	66.0%	0	26.7%	99.9%	51.9%	0	35.5%	99.9%	45.6%	44	5.3%	98.0%	67.0%	0
HE-GENG.D.ENTROPY	6.7%	99.7%	66.0%	0	26.7%	99.9%	51.9%	0	35.5%	99.9%	45.6%	44	5.3%	98.0%	67.0%	0
HE-GENG-LCC.S.MINIMAX	T1				T1				T1				T1			
HE-GENG-LCC.S.ENTROPY	T1				T1				T1				T1			
HE-GENG-LCC.D.MINIMAX	6.7%	99.7%	66.0%	0	26.7%	99.9%	51.9%	0	35.5%	99.9%	45.6%	1	5.3%	98.0%	67.0%	0
HE-GENG-LCC.D.ENTROPY	6.7%	99.7%	66.0%	0	26.7%	99.9%	51.9%	0	35.5%	99.9%	45.6%	1	5.3%	98.0%	67.0%	0
HE-GENG-LN.S.MINIMAX	70.0%	100.0%	21.2%	13	66.7%	100.0%	23.6%	5	64.5%	100.0%	25.1%	11	23.0%	98.6%	54.5%	99
HE-GENG-LN.S.ENTROPY	70.0%	100.0%	21.2%	13	66.7%	100.0%	23.6%	5	64.5%	100.0%	25.1%	11	23.0%	98.6%	54.5%	99
HE-GENG-LN.D.MINIMAX	6.7%	99.7%	66.0%	0	26.7%	99.9%	51.9%	0	35.5%	99.9%	45.6%	2	5.3%	98.0%	67.0%	0
HE-GENG-LN.D.ENTROPY	6.7%	99.7%	66.0%	0	26.7%	99.9%	51.9%	0	35.5%	99.9%	45.6%	2	5.3%	98.0%	67.0%	0
LLC1.THR	0.0%	100.0%	70.7%	328	0.0%	100.0%	70.7%	215	0.0%	100.0%	70.7%	220	0.0%	100.0%	70.7%	804
LLC1.ALPHA	T2				T2				T2				T2			
LLC1.FDR	T2				T2				T2				T2			
LLC2.THR	T1				T1				T1				T1			
LLC2.ALPHA	T2				T2				T2				T2			
LLC2.FDR	T2				T2				T2				T2			
LLC3.THR	T4				T4				T4				T4			
LLC3.BOOTSTRAP	T4				T4				T4				T4			
LLC2-F1.THR	T1				T1				T1				T1			
LLC2-F1.ALPHA	T2				T2				T2				T2			
LLC2-F1.FDR	T2				T2				T2				T2			
LLC2-F2.THR	T1				T1				T1				T1			
LLC2-F2.ALPHA	T2				T2				T2				T2			
LLC2-F2.FDR	T2				T2				T2				T2			
LLC3-F2.THR	T4				T4				T4				T4			
LLC3-F2.BOOTSTRAP	T4				T4				T4				T4			
UNIV-LLC1.THR	0	100.0%	70.7%	328	0.0%	100.0%	70.7%	215	0.0%	100.0%	70.7%	220	0.0%	100.0%	70.7%	804
UNIV-LLC1.ALPHA	86.7%	92.9%	10.7%	328	93.3%	95.3%	5.8%	215	90.3%	95.4%	7.6%	220	90.0%	84.4%	13.1%	804
UNIV-LLC1.FDR	86.7%	92.9%	10.7%	328	93.3%	95.3%	5.8%	215	90.3%	95.4%	7.6%	220	90.0%	84.4%	13.1%	804
UNIV-LLC2.THR	0.0%	100.0%	70.7%	328	0.0%	100.0%	70.7%	215	0.0%	100.0%	70.7%	220	T1			
UNIV-LLC2.ALPHA	70.0%	99.4%	21.2%	328	73.3%	99.7%	18.9%	215	67.7%	99.6%	22.8%	220	T2			
UNIV-LLC2.FDR	53.3%	100.0%	33.0%	328	66.7%	100.0%	23.6%	215	61.3%	100.0%	27.4%	220	T2			
UNIV-LLC3.THR	73.3%	97.8%	18.9%	328	73.3%	98.7%	18.9%	215	83.9%	98.5%	11.5%	220	76.0%	89.5%	18.5%	804
UNIV-LLC3.BOOTSTRAP	0.0%	100.0%	70.7%	328	60.0%	99.1%	28.3%	215	61.3%	99.4%	27.4%	220	0.0%	100.0%	70.7%	804
UNIV-LLC2-F1.THR	3.3%	100.0%	68.4%	10	0.0%	100.0%	70.7%	6	0.0%	100.0%	70.7%	8	T1			
UNIV-LLC2-F1.ALPHA	36.7%	99.6%	44.8%	10	93.3%	97.7%	5.0%	6	90.3%	97.7%	7.0%	8	T2			
UNIV-LLC2-F1.FDR	3.3%	100.0%	68.4%	10	66.7%	99.9%	23.6%	6	61.3%	99.8%	27.4%	8	T2			
UNIV-LLC2-F2.THR	3.33%	100.0%	68.4%	9	0.0%	100.0%	70.7%	7	0.0%	100.0%	70.7%	7	T1			
UNIV-LLC2-F2.ALPHA	63.3%	98.2%	26.0%	9	93.3%	97.6%	5.0%	7	90.3%	97.7%	7.0%	7	T2			
UNIV-LLC2-F2.FDR	6.7%	100.0%	66.0%	9	60.0%	99.8%	28.3%	7	64.5%	99.8%	25.1%	7	T2			
UNIV-LLC3-F2.THR	T4				0.0%	100.0%	70.7%	63	0.0%	100.0%	70.7%	60	T4			
UNIV-LLC3-F2.BOOTSTRAP	T4				26.7%	99.1%	51.9%	63	41.9%	98.6%	41.1%	57	T4			
BIOLEARN.NG	T3				T3				T3				T3			
BIOLEARN.BDE	T3				T3				T3				T3			

Table C3: Detailed results of experiments for YEAST network (4 local causal neighborhoods). See Table C1 for explanation of termination/failure codes T1, T2, T3, and T4.

Appendix D. Assessment of Various Edge Orientation Strategies

To evaluate the accuracy of edge orientation, five orientation methods were tested in two datasets (REGED and ECOLI). All orientation experiments were conducted on the unoriented skeleton discovered by the PC algorithm from observational data. The following five orientation methods were tested:

(1) observational: Edge orientation was determined using constraint-based orientation rules specified in the PC algorithm. This orientation method applied on top of the unoriented PC skeleton is equivalent to the PC algorithm. Notice that some edges may be left unoriented.

(2) experimental: This is a classic orientation approach, and it involves manipulating a variable and assessing its statistical association with the undirected neighbors in order to determine the orientation. For the implementation of this approach, variables with the largest number of undirected neighbors were prioritized for manipulation in order to minimize the number of required experiments (Meganck et al., 2006). Specifically the approach was implemented as follows: (a) Select the vertex with the largest number of undirected neighbors. Denote this variable as X, and its undirected neighbors $Y_1, ..., Y_i, ...Y_n$; (b) Manipulate variable X. (c) For every undirected neighbor Y_i, orient edge as $X \rightarrow Y_i$, if there is a statistically significant association between X and Y_i at $\alpha = 0.05$. Otherwise, orient edge as $Y_i \rightarrow X$; (d) repeat steps (a)-(c) until all edges are oriented.

(3) experimental: For every unoriented edge $X - -Y$in the skeleton, manipulate X and assess the association between X and Y, denoted as A_{XY}. Similarly, manipulate Y and assess the association between X and Y, denoted as A_{YX}. The larger is A_{XY} (or A_{YX}), the stronger is association. If $A_{XY} > A_{YX}$, orient edge as $X \rightarrow Y$, otherwise orient edge as $Y \rightarrow X$;

(4) observational + experimental: apply observational method (1) and orient the rest of the unoriented edges with the experimental method (2);

(5) observational + experimental: apply observational method (1) and orient the rest of the unoriented edges with the experimental method (3).

The results of experiments described above are given in Table D1. The accuracy of orientation is defined as the number of correctly oriented edges divided by the number of correctly inferred edges in the skeleton (i.e. evaluated only with respect to correctly inferred edges by the PC algorithm). In both datasets, the observational orientation had an accuracy that is close to or worse than random (55.2% for REGED and 40.9% for ECOLI). On the other hand, both experimental orientation methods yielded much higher and non-random accuracies up to 100% for REGED dataset and up to 91.2% for ECOLI dataset. Performing observational orientation before experimental orientation reduces the number of experiments as expected, however this also reduces the accuracy. These results indicate that although PC orientation is theoretically sound, experimental orientation methods provide better orientation accuracy.

REGED

Orientation Method	# of edges in the gold-standard	# of edges in the skeleton	# of correctly inferred edges in the skeleton	# of oriented edges in the skeleton	# of correctly inferred edges in the skeleton that are also oriented	# of correctly oriented edges in the skeleton	# of experi-ments	**Accuracy of orientation***
(1) observational	1148	6324	1137	6073	942	520	0	55.2%
(2) experimental	1148	6324	1137	6324	1137	1116	645	98.2%
(3) experimental	1148	6324	1137	6324	1137	1137	1000	100.0%
(4) observational+experimental	1148	6324	1137	6324	1137	712	143	62.6%
(5) observational+experimental	1148	6324	1137	6324	1137	715	336	62.9%

ECOLI

Orientation Method	# of edges in the gold-standard	# of edges in the skeleton	# of correctly inferred edges in the skeleton	# of oriented edges in the skeleton	# of correctly inferred edges in the skeleton that are also oriented	# of correctly oriented edges in the skeleton	# of experi-ments	**Accuracy of orientation***
(1) observational	3632	12091	1660	11964	1595	653	0	40.9%
(2) experimental	3632	12091	1660	12091	1660	1348	1206	81.2%
(3) experimental	3632	12091	1660	12091	1660	1514	1565	91.2%
(4) observational+experimental	3632	12091	1660	12091	1660	718	62	43.3%
(5) observational+experimental	3632	12091	1660	12091	1660	718	152	43.3%

***** Computed only over edges that have been correctly inferred in the skeleton.

Table D1: Comparison of accuracy for various edge orientation methods.

Appendix E. Publicly Available Software Implementations of the Core Methods

Algorithm	Implementation	Link to Publicly Available Software
ODLP*	Matlab	http://ccdlab.org/odlp.html
ALCBN	-	Can be requested from the authors of Meganck et al., 2006
HE-GENG	R	http://www.math.pku.edu.cn:8000/people/view.php?uid=heyb&showdetail=1
LLC	R	LLC1: Can be requested from the authors of Eberhardt et al., 2010 LLC2: https://docs.google.com/file/d/0B7pSUZzmhZ33VnZjdG8xaUVIZDg/edit?pli=1 LLC3: https://docs.google.com/file/d/0B7pSUZzmhZ33b1Zfb3l6XzMwQzQ/edit
BIOLEARN	Java	http://www.c2b2.columbia.edu/danapeerlab/html/biolearn.html

Table E1: Publicly available software implementation of different algorithms

Appendix F. Description of the TIE* and iTIE* algorithms

The TIE* and iTIE* algorithms are described in detail in (Statnikov et al., 2013). Before we review the algorithms below, we note that TIE* and iTIE* were originally introduced for discovery of all Markov boundaries of the target variable T. However, TIE* is also suitable for discovery of all local causal pathways of T consistent with the data when it is used with the Markov boundary induction algorithm Semi-Interleaved HITON-PC; see proof of Theorem 1 in Appendix G for discussion. Similarly, iTIE* which is derived by modifying Semi-Interleaved HITON-PC can be also used for discovery of all local causal pathways of T consistent with the data. When there is no multiplicity of local causal pathways, TIE* and iTIE* will be equivalent to Semi-Interleaved HITON-PC and will output all and only members of the true local causal pathway of T. When the multiplicity is present, the union of Markov boundaries output by TIE* or iTIE* (i.e., all local causal pathways of T consistent with the data) will contain all variables that constitute the true local causal pathway of T and other variables that contain equivalent information about T.

Next, we present the generative TIE* algorithm. This generative algorithm describes a family of related but not identical algorithms which can be seen as instantiations of the same broad algorithmic principles. The pseudo-code of the TIE* generative algorithm is provided in Figure F1. On input TIE* receives (i) a dataset $\mathbb{D}$ (a sample from distribution $\mathbb{P}$) for variables $\boldsymbol{V}$, including a target variable T; (ii) a single Markov boundary induction algorithm $\mathbb{X}$; (iii) a procedure $\mathbb{Y}$ to generate datasets $\mathbb{D}^e$ from the so-called embedded distributions that are obtained by removing subsets of variables from the full set of variables $\boldsymbol{V}$ in the original distribution $\mathbb{P}$; and (iv) a criterion $\mathbb{Z}$ to verify Markov boundaries of T. The inputs $\mathbb{X}, \mathbb{Y}, \mathbb{Z}$ are selected to be suitable for the distribution at hand and should satisfy admissibility rules stated in (Statnikov et al., 2013) for correctness of the algorithm. The algorithm outputs all Markov boundaries of T that exist in the distribution $\mathbb{P}$.

To further facilitate understanding of the TIE* algorithm, we provide in Figure F2 a concrete and specific instantiation of TIE*. Finally, we present in Figure F3 the algorithm iTIE*.

Generative algorithm **TIE***

Inputs:

- dataset $\mathbb{D}$ (a sample from distribution $\mathbb{P}$) for variables ***V***, including a target variable *T*;
- Markov boundary induction algorithm $\mathbb{X}$;
- procedure $\mathbb{Y}$ to generate datasets from the embedded distributions;
- criterion $\mathbb{Z}$ to verify Markov boundaries of *T*.

Output: all Markov boundaries of *T* that exist in $\mathbb{P}$.

1. Use algorithm $\mathbb{X}$ to learn a Markov boundary ***M*** of *T* from the dataset $\mathbb{D}$ for variables ***V*** (i.e., in the original distribution $\mathbb{P}$)
2. Output ***M***
3. Repeat
4. Use procedure $\mathbb{Y}$ to generate a dataset $\mathbb{D}^e$ from the embedded distribution by removing a subset of variables ***G*** from the full set of variables ***V*** in the original distribution (also denoted as $\mathbb{D}(\boldsymbol{V} \setminus \boldsymbol{G})$).
5. Use algorithm $\mathbb{X}$ to learn a Markov boundary $\boldsymbol{M}_{new}$ of *T* from the dataset $\mathbb{D}^e$
6. If $\boldsymbol{M}_{new}$ is a Markov boundary of *T* in the original distribution according to criterion $\mathbb{Z}$, output $\boldsymbol{M}_{new}$
7. Until all datasets $\mathbb{D}^e$ generated by procedure $\mathbb{Y}$ have been considered.

Figure F1: TIE* generative algorithm

An example of instantiated algorithm **TIE***

Inputs: dataset $\mathbb{D}$ (a sample from distribution $\mathbb{P}$) for variables ***V***, including a target variable *T*.

Output: all Markov boundaries of *T* that exist in $\mathbb{P}$.

1. Use algorithm Semi-Interleaved HITON-PC to learn a Markov boundary ***M*** of *T* from the dataset $\mathbb{D}$ for variables ***V*** (i.e., in the original distribution $\mathbb{P}$)
2. Output ***M***
3. Repeat
4. Generate a dataset $\mathbb{D}^e = \mathbb{D}(\boldsymbol{V} \setminus \boldsymbol{G})$ from the embedded distribution by removing from the full set of variables ***V*** in the original distribution the smallest subset ***G*** of the so far discovered Markov boundaries of *T* such that:
   (i) ***G*** was not considered in the previous iterations of this step, and
   (ii) ***G*** does not include any subset of variables that was previously removed from ***V*** to yield a dataset $\mathbb{D}^e$ when $\boldsymbol{M}_{new}$ was found not to be a Markov boundary of *T* in the original distribution (per step 6)
5. Use algorithm Semi-Interleaved HITON-PC to learn a Markov boundary $\boldsymbol{M}_{new}$ of *T* from the dataset $\mathbb{D}^e$ (i.e., in the embedded distribution)
6. If $T \perp \boldsymbol{M} \mid \boldsymbol{M}_{new}$, then $\boldsymbol{M}_{new}$ is a Markov boundary of *T* in the original distribution and it is output by the algorithm
7. Until all datasets $\mathbb{D}^e$ generated in step 4 have been considered.

Figure F2: An example of instantiated TIE* algorithm.

Algorithm **iTIE***

Input: dataset $\mathbb{D}$ (a sample from distribution $\mathbb{P}$) for variables ***V***, including a target variable *T*.

Output: multiple Markov boundaries of *T* that exist in $\mathbb{P}$.

Phase I: Forward

1. Initialize Θ with an empty set
2. Initialize ***M*** with an empty set
3. Initialize the set of eligible variables ***E*** ← ***V*** \ *T*
4. Repeat
5. *Y* ← $\text{argmax}_{X \in \boldsymbol{E}}$ Association(*T*, *X*)
6. ***E*** ← ***E*** \ *Y*
7. If there is no subset ***Z*** ⊂ ***M*** such that $T \perp Y \mid \boldsymbol{Z}$ then
8. ***M*** ← ***M*** ∪ *Y*
9. Else if ***Z*** exists and the following relations hold: $T \not\perp Y$, $T \not\perp \boldsymbol{Z}$, $T \perp \boldsymbol{Z} \mid Y$
10. Record in Θ that *Y* and ***Z*** contain equivalent information with respect to *T*
11. Until ***E*** is empty

Phase II: Backward

12. For each *X* ∈ ***M***
13. If there is a subset ***Z*** ⊂ ***M*** \ *X* such that $T \perp X \mid \boldsymbol{Z}$ then
14. ***M*** ← ***M*** \ *X*

Phase III: Construction of multiple Markov boundaries

15. Compute the Cartesian product of target information equivalency relations for subsets of ***M*** that are stored in Θ to construct multiple Markov boundaries of *T*
16. Output multiple Markov boundaries of *T*

Figure F3: iTIE* algorithm

Appendix G. Proof of Correctness of ODLP

Theorem 1 *ODLP is sound under the following sufficient assumptions: (i) TIE near-faithfulness (as a relaxation of local adjacency faithfulness to allow for target information equivalency relations); (ii) causal Markov condition; (iii) local causal sufficiency; (iv) acyclicity of the data-generative graph; and (v) correctness of statistical decisions.*

Proof First, we remind the readers that under DAG-faithfulness, the Markov boundary is unique and consists of children, parents, and spouses of T. i.e., the Markov boundary contains all members of the local causal pathway of T (consisting of parents and children of T), plus spouses that are not children of T. The latter spouses are marginally or conditionally independent of T unlike members of the local causal pathways of T. Under DAG-faithfulness, the Semi-Interleaved HITON-PC algorithm can discover all members of the local causal pathway of T (Aliferis et al., 2010a,b). However under TIE near-faithfulness, this algorithm will output a local causal pathway consistent with the data, which may or may not contain parents and children of T.

We have previously established that an admissible instantiation of the generative algorithm TIE* can correctly discover all Markov boundaries of the target variable T (see Theorem 10 in (Statnikov et al., 2013)). When TIE* is instantiated with the Markov boundary inducer Semi-Interleaved HITON-PC, it will identify in step 1 all local causal pathways of T consistent with the data (Statnikov et al., 2013). The latter requires that members of all local causal pathways consistent with the data are marginally and conditionally dependent on T (except for violations of the intersection property that lead to equivalence relations), which is satisfied given assumptions of this theorem, in particular TIE near-faithfulness. Therefore, all members of the true local causal pathway will be contained in the output of TIE* in step 1.

Similarly, it can be shown that iTIE* will identify in step 1 all local causal pathways consistent with the data (and therefore all members of the true local causal pathway) given assumptions of this theorem and an additional requirement that all equivalence relations in the underlying distribution follow from equivalence relations of individual variables. The latter requirement is one of sufficient assumptions for iTIE* correctness (Statnikov et al., 2013).

Before we proceed with the remainder of the proof, we examine the contents of equivalence clusters formed in step 3. Given three types of variables of interest (causes, effects, and passengers) there are the following options for contents of the cluster: (1) causes; (2) causes and effects; (3) causes and passengers; (4) causes, effects and passengers; (5) effects; (6) effects and passengers; and (7) passengers. It can be shown by examples that options (1)-(5) are possible and consistent with assumptions of this theorem. On the other hand, options (6) and (7) cannot take place in the settings of this theorem.

Next we prove correctness of identification of effects, direct effects, other effects ("other effects" are effects that are not identified as direct effects, they could be indirect effects or both direct and indirect effects at the same time), causes, direct causes, other causes ("other causes" are causes that are not identified as direct causes, they could be indirect causes or both direct and indirect causes at the same time), and passengers within the variable set $\boldsymbol{V}$, which is the union of all variables that participate in the local causal pathways

of T consistent with the data. Given that all members of the true local causal pathway are contained in the set $\boldsymbol{V}$, the correct identification of direct effects and direct causes within the set $\boldsymbol{V}$ implies that ODLP is sound.

Identification of effects and direct/other effects: Based on the assumption of correctness of statistical decisions and the definition of causation, all effects of T are correctly identified by performing an experiment on T (step 4) and considering as effects all variables $\boldsymbol{E} \subseteq \boldsymbol{V}$ that change as a result of that experiment (step 5). Identification of direct/other effects is performed within the subset E. We distinguish here three cases:

1. An equivalence cluster contains one variable X, which is an effect (step 9.a). Then X has to be a direct effect. Otherwise, based on causal Markov condition and correctness of statistical decisions, X will not belong to $\boldsymbol{E}$ because X will be rendered statistically independent of T conditioned on a subset of variables from any local causal pathway of T consistent with the data during execution of TIE* in step 1.

2. An equivalence cluster contains multiple variables, out of which only one variable X (effect) has neither been identified yet as other effect nor as direct effect and all other effect variables have been identified as other effects (step 9.a). Then, similarly to the previous case, X has to be a direct effect. Otherwise, a cluster will only have an indirect but no direct effect which cannot happen based on the assumptions of this theorem and the methodology of constructing equivalence clusters by utilizing TIE* in steps 1-3.

3. An equivalence cluster contains multiple variables, out of which two or more effect variables have neither been identified as other effects nor as direct effects. The algorithm proceeds to execution of steps 9.b-9.d, whose correctness follows from the definition of causation and the assumption of correctness of statistical decisions.

Identification of causes and direct/other causes: Since we have already identified the set of effects $\boldsymbol{E}$, identification of causes (and direct/other causes) is performed within the set of variables $\boldsymbol{V} \setminus \boldsymbol{E}$. We distinguish here three cases:

1. An equivalence cluster contains one unmarked variable X (step 6.a). Since X is unmarked, it is not an effect. Then X has to be a direct cause. Otherwise, based on causal Markov condition and correctness of statistical decisions, X will not belong to $\boldsymbol{V} \setminus \boldsymbol{E}$ because X will be rendered statistically independent of T conditioned on a subset of variables from any local causal pathway of T consistent with the data during execution of TIE* in step 1.

2. An equivalence cluster contains multiple variables, out of which only one variable X has not been marked yet and all other variables have been identified as passengers and/or effects (step 6.a). Again, since X is unmarked, it is not an effect. Then, similarly to the previous case, X has to be a direct cause. Otherwise, a cluster will either have only effects and passengers or effects, passengers, and an indirect cause. None of these cases can happen based on the assumptions of this theorem and the methodology of constructing equivalence clusters by utilizing TIE* in steps 1-3.

3. An equivalence cluster contains multiple variables, out of which two or more variables have not been marked yet. The algorithm proceeds to execution of steps 6.b-6.d, whose correctness follows from the definition of causation and the assumption of correctness of statistical decisions.

Identification of passengers: Based on the assumption of correctness of statistical decisions and the definition of causation, passengers are correctly identified in step 6.d. More

specifically, all variables marked as passengers in that step have been previously unmarked (and therefore are not effects of T) and are not on the causal path to T (and therefore are not causes of T). ■

Appendix H. More on ODLP's Experimental Strategy and its Efficiency

Consider an example network shown in Figure H1.a. Variables A, B, C, D, and E contain equivalent information about the target variable T and cannot be distinguished with observational data. Without any prior knowledge about the causal role of A, B, C, D, and E, we will first need to manipulate T to determine that none of the above 5 variables is an effect of T. Therefore, they can be either causes or passengers. If we manipulate C, we will realize that D and E change but T does not change due to manipulation of C. Therefore, C, D, E are all passengers and we do not need to manipulate D and E (we saved 2 experiments). Next we manipulate A and observe that it leads to changes in T (and B, C, D, and E) and thus it is a cause of T. Finally, we can manipulate B and observe that it leads to changes only in T and thus it is a direct cause. So, in total we performed 4 experiments (manipulate T, C, A, and B in order). However, if we did not choose C early on for manipulations, we could end up doing up to 6 experiments (manipulate T, E, D, C, A, and B in order) to identify the local causal pathway. In fact, it is not possible to conduct fewer than four single-variable experiments in this example, and thus the sequence of experiments T, C, A, B is optimal. The only problem is that we do not know the graphical structure when we perform experiments, and thus we need to resort to heuristics to manipulate first variables that are likely to yield savings in experiments (step 6.b of the ODLP algorithm; see Figure 3).

Consider another example network shown in Figure H1.b. Variables A, B, C, D, E, F, and J contain equivalent information about the target variable T and cannot be distinguished with observational data. Assume that we have already manipulated T, A, and B, and now we are deciding what variable to manipulate next. Manipulation of T, A, and B provided us with partial information on topological (causal) order of variables. Specifically, we know that (i) no variable is downstream of T (from manipulating T), (ii) B, C, D, E, F, J, and T are downstream of A (from manipulating A), and (iii) D, E, F, J, and T are downstream of B (from manipulating B). As discussed in the text, one possibility is to use a partial network-based heuristic that chooses a variable that has the highest topological order relative to T. As established from constraints learned from experimental data, variable C has the highest topological order and has not been manipulated yet. Manipulation of C allows to immediately identify the local causal pathway because D, E, F, and J will change and T will not change due to manipulation of C, thus C, D, E, F, and J are all passengers. In summary we conducted 4 experiments, while alternative strategies will take up to 8 experiments. To see the expected efficiency of the above heuristic function, we can revisit this example and assume that we do not have knowledge to manipulate A and B first. In this case, we will identify the local causal pathway in 4 experiments with probability 6.67% using the above heuristic and with probability 2.86% without the heuristic and performing random selection of variables for manipulation (in step 6.b of the ODLP algorithm; see Figure 3).

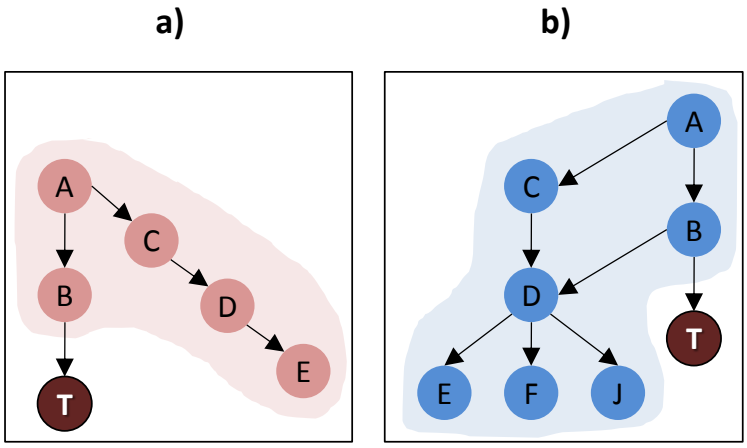


Figure H1: Two causal networks used to illustrate ODLPs experimental strategy and its efficiency. Variables are shown with circles, and edges represent direct causal influences. The target variable is T. Variables that are shown with the same color contain the same information about the target (they are target information equivalent).

References

Constantin F Aliferis, Alexander Statnikov, Ioannis Tsamardinos, Subramani Mani, and Xenofon D Koutsoukos. Local causal and markov blanket induction for causal discovery and feature selection for classification part i: Algorithms and empirical evaluation. *The Journal of Machine Learning Research*, 11:171–234, 2010a.

Constantin F Aliferis, Alexander Statnikov, Ioannis Tsamardinos, Subramani Mani, and Xenofon D Koutsoukos. Local causal and markov blanket induction for causal discovery and feature selection for classification part ii: Analysis and extensions. *The Journal of Machine Learning Research*, 11:235–284, 2010b.

Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 289–300, 1995.

Yoav Benjamini and Daniel Yekutieli. The control of the false discovery rate in multiple testing under dependency. *Annals of Statistics*, pages 1165–1188, 2001.

David Maxwell Chickering. Optimal structure identification with greedy search. *The Journal of Machine Learning Research*, 3:507–554, 2003.

Diego Colombo and Marloes H Maathuis. Order-independent constraint-based causal structure learning. *The Journal of Machine Learning Research*, 15(1):3741–3782, 2014.

Povilas Daniusis, Dominik Janzing, Joris Mooij, Jakob Zscheischler, Bastian Steudel, Kun Zhang, and Bernhard Schölkopf. Inferring deterministic causal relations. *arXiv preprint arXiv:1203.3475*, 2012.

Edward R Dougherty and Marcel Brun. On the number of close-to-optimal feature sets. *Cancer Informatics*, 2:189, 2006.

Frederick Eberhardt, Patrik O Hoyer, and Richard Scheines. Combining experiments to discover linear cyclic models with latent variables. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 185–192, 2010.

Clark N. Glymour and Gregory F. Cooper. *Computation, Causation, and Discovery.* AAAI Press ; MIT Press, Menlo Park, California, 1999.

Isabelle Guyon, Constantin F Aliferis, Gregory F Cooper, André Elisseeff, Jean-Philippe Pellet, Peter Spirtes, and Alexander R Statnikov. Design and analysis of the causation and prediction challenge. In *WCCI Causation and Prediction Challenge*, pages 1–33, 2008.

Yang-Bo He and Zhi Geng. Active learning of causal networks with intervention experiments and optimal designs. *Journal of Machine Learning Research*, 9(11), 2008.

Patrik O Hoyer, Dominik Janzing, Joris M Mooij, Jonas Peters, and Bernhard Schölkopf. Nonlinear causal discovery with additive noise models. In *Advances in Neural Information Processing Systems*, pages 689–696, 2009.

Antti Hyttinen, Frederick Eberhardt, and Patrik O Hoyer. Causal discovery for linear cyclic models with latent variables. *on Probabilistic Graphical Models*, page 153, 2010.

Antti Hyttinen, Frederick Eberhardt, and Patrik O Hoyer. Learning linear cyclic causal models with latent variables. *The Journal of Machine Learning Research*, 13(1):3387–3439, 2012.

Dominik Janzing, Joris Mooij, Kun Zhang, Jan Lemeire, Jakob Zscheischler, Povilas Daniušis, Bastian Steudel, and Bernhard Schölkopf. Information-geometric approach to inferring causal directions. *Artificial Intelligence*, 182:1–31, 2012.

Jan Lemeire. *Learning Causal Models of Multivariate Systems and the Value of it for the Performance Modeling of Computer Programs.* PhD thesis, Vrije Universiteit Brussel, 2007.

Daniel Marbach, Thomas Schaffter, Claudio Mattiussi, and Dario Floreano. Generating realistic in silico gene networks for performance assessment of reverse engineering methods. *Journal of Computational Biology*, 16(2):229–239, 2009.

Stijn Meganck, Philippe Leray, and Bernard Manderick. Learning causal bayesian networks from observations and experiments: A decision theoretic approach. In *Proceedings of the Third International Conference on Modeling Decisions for Artificial Intelligence*, MDAI'06, pages 58–69, Berlin, Heidelberg, 2006. Springer-Verlag.

Joshua Menke and Tony R Martinez. Using permutations instead of student's t distribution for p-values in paired-difference algorithm comparisons. In *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*, volume 2, pages 1331–1335. IEEE, 2004.

Joris Mooij, Oliver Stegle, Dominik Janzing, Kun Zhang, and Bernhard Schölkopf. Probabilistic latent variable models for distinguishing between cause and effect. In *Advances in Neural Information Processing Systems*, pages 1687–1695, 2010.

Kevin P Murphy. Active learning of causal bayes net structure, 2001.

Richard E. Neapolitan. *Learning Bayesian Networks*. Prentice Hall, illustrated edition edition, April 2003. ISBN 0130125342.

Judea Pearl. *Probabilistic Reasoning in Intelligent Systems : Networks of Plausible Inference*. Morgan Kaufmann Publishers, 1 edition, September 1997. ISBN 9781558604797.

Judea Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, New York, NY, USA, 2nd edition, 2009. ISBN 052189560X, 9780521895606.

Dana Pe'er, Aviv Regev, Gal Elidan, and Nir Friedman. Inferring subnetworks from perturbed expression profiles. *Bioinformatics*, 17, 2001.

Jose M Peña, Roland Nilsson, Johan Björkegren, and Jesper Tegnér. Towards scalable and data efficient learning of markov boundaries. *International Journal of Approximate Reasoning*, 45(2):211–232, 2007.

Joseph Ramsey, Jiji Zhang, and Peter L Spirtes. Adjacency-faithfulness and conservative causal inference. In *Proceedings of the 22nd Annual Conference on Uncertainty in Artificial Intelligence (UAI-2006)*, pages 401–408, 2006.

Thomas Richardson. A discovery algorithm for directed cyclic graphs. In *Proceedings of the Twelfth International Conference on Uncertainty in Artificial Intelligence*, pages 454–461. Morgan Kaufmann Publishers Inc., 1996.

Thomas Richardson and Peter Spirtes. *Automated Discovery of Linear Feedback Models*, chapter 7, pages 254–302. MIT Press, 1999.

Karen Sachs, Omar Perez, Dana Pe'er, Douglas A Lauffenburger, and Garry P Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308(5721):523–529, 2005.

Thomas Schaffter, Daniel Marbach, and Dario Floreano. Genenetweaver: in silico benchmark generation and performance profiling of network inference methods. *Bioinformatics*, 27(16):2263–2270, 2011.

Shohei Shimizu, Patrik O Hoyer, Aapo Hyvärinen, and Antti Kerminen. A linear non-gaussian acyclic model for causal discovery. *The Journal of Machine Learning Research*, 7:2003–2030, 2006.

Peter Spirtes, Clark N Glymour, and Richard Scheines. *Causation, Prediction, and Search*, volume 81. MIT press, 2000.

Alexander Statnikov and Constantin F Aliferis. Analysis and computational dissection of molecular signature multiplicity. *PLoS Computational Biology*, 6(5):e1000790, 2010.

Alexander Statnikov, Ioannis Tsamardinos, Laura E Brown, and Constantin F Aliferis. *Causal explorer: A matlab library of algorithms for causal discovery and variable selection for classification*, volume 2, page 267. Microtome Publishing, 2010.

Alexander Statnikov, Mikael Henaff, Nikita I Lytkin, and Constantin F Aliferis. New methods for separating causes from effects in genomics data. *BMC Genomics*, 13, 2012.

Alexander Statnikov, Jan Lemeir, and Constantin F Aliferis. Algorithms for discovery of multiple markov boundaries. *The Journal of Machine Learning Research*, 14(1):499–566, 2013.

Simon Tong and Daphne Koller. Active learning for structure in bayesian networks. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'01, pages 863–869, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

Ioannis Tsamardinos, Laura E Brown, and Constantin F Aliferis. The max-min hill-climbing bayesian network structure learning algorithm. *Machine Learning*, 65(1):31–78, 2006a.

Ioannis Tsamardinos, Alexander R Statnikov, Laura E Brown, and Constantin F Aliferis. Generating realistic large bayesian networks by tiling. In *FLAIRS Conference*, pages 592–597, 2006b.

Jianxin Yin, You Zhou, Changzhang Wang, Ping He, Cheng Zheng, and Zhi Geng. Partial orientation and local structural learning of causal networks for prediction. In *WCCI Causation and Prediction Challenge*, pages 93–105, 2008.

Kun Zhang and Aapo Hyvärinen. Distinguishing causes from effects using nonlinear acyclic causal models. In *Journal of Machine Learning Research, Workshop and Conference Proceedings (NIPS 2008 causality workshop)*, volume 6, pages 157–164, 2008.

Journal of Machine Learning Research 16 (2015) 3269-3297 Submitted 1/15; Published 12/15

Plug-and-Play Dual-Tree Algorithm Runtime Analysis

Ryan R. Curtin RYAN@RATML.ORG
School of Computational Science and Engineering
Georgia Institute of Technology
Atlanta, GA 30332-0250, USA

Dongryeol Lee DRSELEE@GMAIL.COM
Yahoo Labs
Sunnyvale, CA 94089

William B. March MARCH@ICES.UTEXAS.EDU
Institute for Computational Engineering and Sciences
University of Texas, Austin
Austin, TX 78712-1229

Parikshit Ram P.RAM@GATECH.EDU
Skytree, Inc.
Atlanta, GA 30332

Editor: Nando de Freitas

Abstract

Numerous machine learning algorithms contain pairwise statistical problems at their core—that is, tasks that require computations over all pairs of input points if implemented naively. Often, tree structures are used to solve these problems efficiently. Dual-tree algorithms can efficiently solve or approximate many of these problems. Using cover trees, rigorous worst-case runtime guarantees have been proven for some of these algorithms. In this paper, we present a *problem-independent* runtime guarantee for *any* dual-tree algorithm using the cover tree, separating out the problem-dependent and the problem-independent elements. This allows us to just plug in bounds for the problem-dependent elements to get runtime guarantees for dual-tree algorithms for any pairwise statistical problem without re-deriving the entire proof. We demonstrate this plug-and-play procedure for nearest-neighbor search and approximate kernel density estimation to get improved runtime guarantees. Under mild assumptions, we also present the first linear runtime guarantee for dual-tree based range search.

Keywords: dual-tree algorithms, adaptive runtime analysis, cover tree, expansion constant, nearest neighbor search, kernel density estimation, range search

1. Dual-tree Algorithms

A surprising number of machine learning algorithms have computational bottlenecks that can be expressed as pairwise statistical problems. By this, we mean computational tasks that can be evaluated directly by iterating over all pairs of input points. Nearest neighbor search is one such problem, since for every query point, we can evaluate its distance to every reference point and keep the closest one. This naively requires $O(N)$ time to answer a single query in a reference set of size N; answering $O(N)$ queries subsequently requires

prohibitive $O(N^2)$ time. Kernel density estimation is also a pairwise statistical problem, since we compute a sum over all reference points for each query point. This again requires $O(N^2)$ time to answer $O(N)$ queries if done directly. The reference set is typically indexed with spatial data structures to accelerate this type of computation (Finkel and Bentley, 1974; Beygelzimer et al., 2006); these result in $O(\log N)$ runtime per query under favorable conditions.

Building upon this intuition, Gray and Moore (2001) generalized the fast multipole method from computational physics to obtain dual-tree algorithms. These are extremely useful when there are large query sets, not just a few query points. Instead of building a tree on the reference set and searching with each query point separately, Gray and Moore suggest also building a query tree and traversing both the query and reference trees simultaneously (a *dual-tree traversal*, from which the class of algorithms takes its name).

Dual-tree algorithms can be easily understood through the recent framework of Curtin et al. (2013b): two trees (a query tree and a reference tree) are traversed by a *pruning dual-tree traversal*. This traversal visits combinations of nodes from the trees in some sequence (each combination consisting of a query node and a reference node), calling a problem-specific `Score()` function to determine if the node combination can be pruned. If not, then a problem-specific `BaseCase()` function is called for each combination of points held in the query node and reference node. This has significant similarity to the more common single-tree branch-and-bound algorithms, except that the algorithm must recurse into child nodes of *both* the query tree and reference tree.

There exist numerous dual-tree algorithms for problems as diverse as kernel density estimation (Gray and Moore, 2003), mean shift (Wang et al., 2007), minimum spanning tree calculation (March et al., 2010), n-point correlation function estimation (March et al., 2012), max-kernel search (Curtin et al., 2013c), particle smoothing (Klaas et al., 2006), variational inference (Amizadeh et al., 2012), range search (Gray and Moore, 2001), and embedding techniques (Van Der Maaten, 2014), to name a few.

Some of these algorithms are derived using the cover tree (Beygelzimer et al., 2006), a data structure with compelling theoretical qualities. When cover trees are used, dual-tree all-nearest-neighbor search and approximate kernel density estimation have $O(N)$ runtime guarantees for $O(N)$ queries (Ram et al., 2009a); minimum spanning tree calculation scales as $O(N \log N)$ (March et al., 2010). Other problems have similar worst-case guarantees (Curtin and Ram, 2014; March, 2013).

In this work we combine the generalization of Curtin et al. (2013b) with the theoretical results of Beygelzimer et al. (2006) and others in order to develop a worst-case runtime bound for any dual-tree algorithm when the cover tree is used.

Section 2 lays out the required background, notation, and introduces the cover tree and its associated theoretical properties. Readers familiar with the cover tree literature and dual-tree algorithms (especially Curtin et al., 2013b) may find that section to be review. Following that, we introduce an intuitive measure of cover tree imbalance, an important property for understanding the runtime of dual-tree algorithms, in Section 3. This measure of imbalance is then used to prove the main result of the paper in Section 4, which is a worst-case runtime bound for generalized dual-tree algorithms. We apply this result to three specific problems: nearest neighbor search (Section 5), approximate kernel density estimation (Section 6), and range search / range count (Section 7), showing linear runtime

Symbol	Description
$\mathscr{N}$	A tree node
$\mathscr{C}_i$	Set of child nodes of $\mathscr{N}_i$
$\mathscr{P}_i$	Set of points held in $\mathscr{N}_i$
$\mathscr{D}_i^n$	Set of descendant nodes of $\mathscr{N}_i$
$\mathscr{D}_i^p$	Set of points contained in $\mathscr{N}_i$ and $\mathscr{D}_i^n$
μ_i	Center of $\mathscr{N}_i$
λ_i	Furthest descendant distance from μ_i

Table 1: Notation for trees. See Curtin et al. (2013b) for details.

bounds for each of those algorithms. Each of these bounds is an improvement on the state-of-the-art, and in the case of range search, is the first such bound. Despite the intuition this provides for the scaling properties of all dual-tree algorithms[1], it must be kept in mind that these worst-case bounds only apply to dual-tree algorithms that use the cover tree and the standard cover tree traversal.

2. Preliminaries

For simplicity, the algorithms considered in this paper will be presented in a tree-independent context, as in Curtin et al. (2013b), but the only type of tree we will consider is the cover tree (Beygelzimer et al., 2006), and the only type of traversal we will consider is the cover tree pruning dual-tree traversal, which we will describe later.

As we will be making heavy use of trees, we must establish notation (taken from Curtin et al., 2013b). The notation we will be using is defined in Table 1.

2.1 The Cover Tree

The cover tree is a leveled hierarchical data structure originally proposed for the task of nearest neighbor search by Beygelzimer et al. (2006). Each node $\mathscr{N}_i$ in the cover tree is associated with a single point p_i. An adequate description is given in their work (we have adapted notation slightly):

> A *cover tree* $\mathscr{T}$ on a dataset S is a leveled tree where each level is a "cover" for the level beneath it. Each level is indexed by an integer scale s_i which decreases as the tree is descended. Every *node* in the tree is associated with a point in S. Each *point* in S may be associated with multiple nodes in the tree; however, we require that any point appears at most once in every level. Let C_{s_i} denote the set of points in S associated with the nodes at level s_i. The cover tree obeys the following invariants for all s_i:

1. Dual-tree algorithms using *kd*-trees and other types of trees have been observed to empirically scale linearly for tasks that take quadratic time without the use of trees; see the empirical results of Gray and Moore (2001); March et al. (2010); Vladymyrov and Carreira-Perpinán (2014); Klaas et al. (2006); Gray and Moore (2003).

- *(Nesting).* $C_{s_i} \subset C_{s_i-1}$. This implies that once a point $p \in S$ appears in C_{s_i} then *every* lower level in the tree has a node associated with p.
- *(Covering tree).* For every $p_i \in C_{s_i-1}$, there exists a $p_j \in C_{s_i}$ such that $d(p_i, p_j) < 2^{s_i}$ and the node in level s_i associated with p_j is a parent of the node in level $s_i - 1$ associated with p_i.
- *(Separation).* For all distinct $p_i, p_j \in C_{s_i}$, $d(p_i, p_j) > 2^{s_i}$.

As a consequence of this definition, if there exists a node $\mathscr{N}_i$, containing the point p_i at some scale s_i, then there will also exist a self-child node $\mathscr{N}_{ic}$ containing the point p_i at scale $s_i - 1$ which is a child of $\mathscr{N}_i$. In addition, every descendant point of the node $\mathscr{N}_i$ is contained within a ball of radius 2^{s_i+1} centered at the point p_i; therefore, $\lambda_i = 2^{s_i+1}$ and $\mu_i = p_i$ (Table 1).

Note that the cover tree may be interpreted as an infinite-leveled tree, with C_∞ containing only the root point, $C_{-\infty} = S$, and all levels between defined as above. Beygelzimer et al. (2006) find this representation (which they call the *implicit* representation) easier for description of their algorithms and some of their proofs. But clearly, this is not suitable for implementation; hence, there is an *explicit* representation in which all nodes that have only a self-child are coalesced upwards (that is, the node's self-child is removed, and the children of that self-child are taken to be the children of the node). Figure 1 shows each of the levels of an example cover tree (in the explicit representation) on a simple six-point dataset.

In this work, we consider only the explicit representation of a cover tree, and do not concern ourselves with the details of tree construction[2].

2.2 Expansion Constant

The explicit representation of a cover tree has a number of useful theoretical properties based on the expansion constant (Karger and Ruhl, 2002); we restate its definition below.

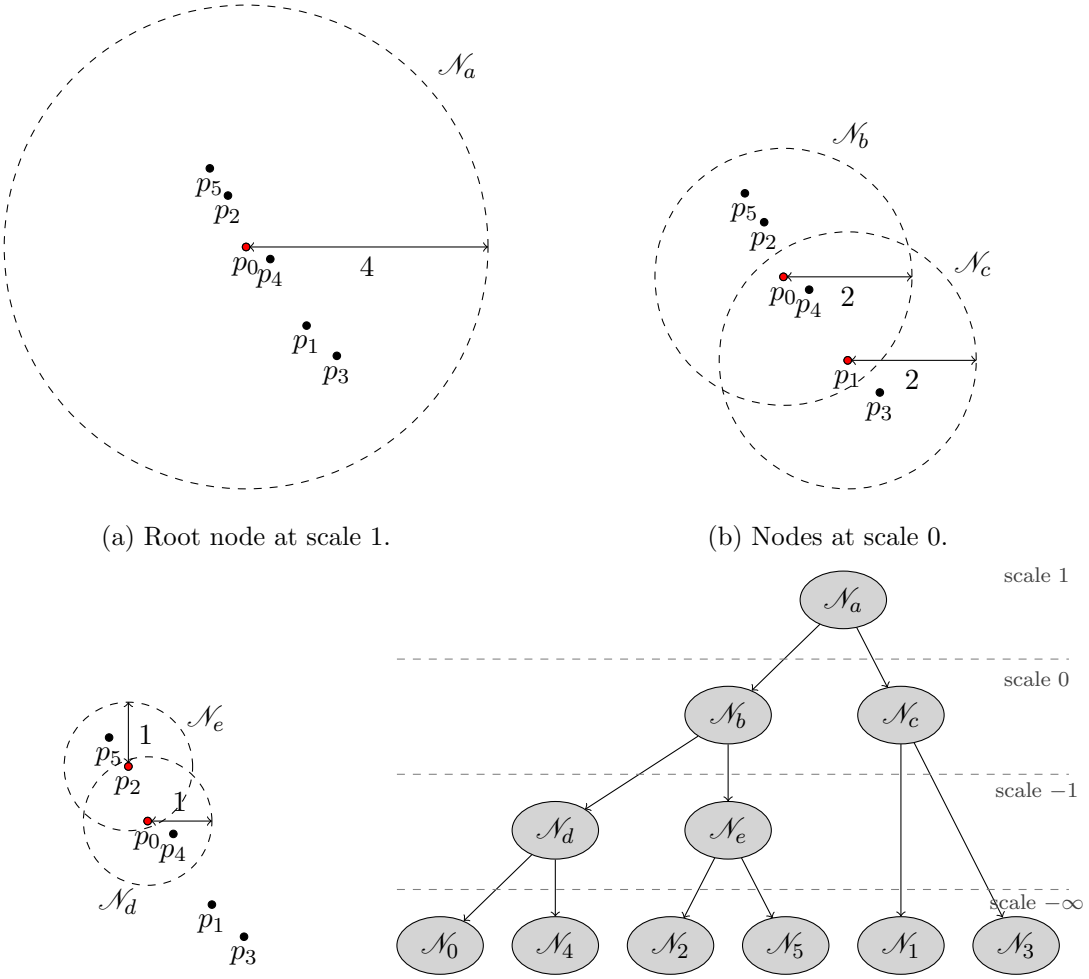
Definition 1 *Let $B_S(p, \Delta)$ be the set of points in S within a closed ball of radius Δ around some $p \in S$ with respect to a metric d: $B_S(p, \Delta) = \{r \in S : d(p, r) \leq \Delta\}$. Then, the* **expansion constant** *of S with respect to the metric d is the smallest $c \geq 2$ such that*

$$|B_S(p, 2\Delta)| \leq c|B_S(p, \Delta)| \;\forall\; p \in S,\; \forall\; \Delta > 0. \tag{1}$$

The expansion constant is used heavily in the cover tree literature. It is, in some sense, a notion of instrinic dimensionality, most useful in scenarios where c is independent of the number of points in the dataset (Karger and Ruhl, 2002; Beygelzimer et al., 2006; Krauthgamer and Lee, 2004; Ram et al., 2009a). Note also that if points in $S \subset \mathcal{H}$ are being drawn according to a stationary distribution $f(x)$, then c will converge to some finite value c_f as $|S| \to \infty$. To see this, define c_f as a generalization of the expansion constant for distributions. $c_f \geq 2$ is the smallest value such that

$$\int_{\mathcal{B}_{\mathcal{H}}(p,2\Delta)} f(x)dx \leq c_f \int_{\mathcal{B}_{\mathcal{H}}(p,\Delta)} f(x)dx \tag{2}$$

2. A batch construction algorithm is given by Beygelzimer et al. (2006), called `Construct`.



(a) Root node at scale 1.

(b) Nodes at scale 0.

(c) Nodes at scale −1.

(d) Abstract representation.

Figure 1: Example cover tree on six points in $\mathcal{R}^2$. (a) $\mathcal{N}_a$ is centered at p_0 with scale 1. (b) $\mathcal{N}_b$ and $\mathcal{N}_c$ are centered at p_0 and p_1, respectively, and have scale 0. (c) $\mathcal{N}_d$ and $\mathcal{N}_e$ are centered at p_0 and p_2, respectively, and have scale -1. The leaves, $\mathcal{N}_0$ through $\mathcal{N}_6$, are centered at each of the six points, with scale $-\infty$ (and therefore radius 0). Note that although node $\mathcal{N}_b$ in subfigure (b) overlaps node $\mathcal{N}_c$, point p_1 only belongs to $\mathcal{N}_c$, not $\mathcal{N}_b$. Note also that this is only one valid cover tree that could be built on the data; other configurations are possible; for instance, selecting a different root point gives different valid cover trees.

for all $p \in \mathcal{H}$ and $\Delta > 0$ such that $\int_{\mathcal{B}_{\mathcal{H}}(p,\Delta)} f(x)dx > 0$, and with $\mathcal{B}_{\mathcal{H}}(p, \Delta)$ defined as the closed ball of radius Δ in the space $\mathcal{H}$.

As a simple example, take $f(x)$ as a uniform spherical distribution in $\mathcal{R}^d$: for any $|x| \leq 1$, $f(x)$ is a constant; for $|x| > 1$, $f(x) = 0$. It is easy to see that c_f in this situation is 2^d, and thus for some dataset S, c must converge to that value as more and more points are added to S. Closed-form solutions for c_f for more complex distributions are less easy to derive; however, empirical speedup results from Beygelzimer et al. (2006) suggest the existence of datasets where c is not strongly dependent on d. For instance, the `covtype` dataset has 54 dimensions but the expansion constant is much smaller than other, lower-dimensional datasets.

There are some other important observations about the behavior of c. Adding a single point to S may increase c arbitrarily: consider a set S distributed entirely on the surface of a unit hypersphere. If one adds a single point at the origin, producing the set S', then c explodes to $|S'|$ whereas before it may have been much smaller than $|S|$. Adding a single point may also decrease c significantly. Suppose one adds a point arbitrarily close to the origin to S'; now, the expansion constant will be $|S'|/2$. Both of these situations are degenerate cases not commonly encountered in real-world behavior; we discuss them in order to point out that although we can bound the behavior of c as $|S| \to \infty$ for S from a stationary distribution, we are not able to easily say much about its convergence behavior.

The expansion constant can be used to show a few useful bounds on various properties of the cover tree; we restate these results below, given some cover tree built on a dataset S with expansion constant c and $|S| = N$:

- **Width bound:** no cover tree node has more than c^4 children (Lemma 4.1, Beygelzimer et al., 2006).
- **Depth bound:** the maximum depth of any node is $O(c^2 \log N)$ (Lemma 4.3, Beygelzimer et al., 2006).
- **Space bound:** a cover tree has $O(N)$ nodes (Theorem 1, Beygelzimer et al., 2006).

Lastly, we introduce a convenience lemma of our own which is a generalization of the packing arguments used by Beygelzimer et al. (2006). This is a more flexible version of their argument.

Lemma 1 *Consider a dataset S with expansion constant c and a subset $C \subseteq S$ such that every two distinct points in C are separated by at least δ. Then, for any point p (which may or may not be in S), and any radius $\rho\delta > 0$:*

$$|B_S(p, \rho\delta) \cap C| \leq c^{2+\lceil \log_2 \rho \rceil}. \tag{3}$$

Proof The proof is based on the packing argument from Lemma 4.1 in Beygelzimer et al. (2006). Consider two cases: first, let $d(p, p_i) > \rho\delta$ for any $p_i \in S$. In this case, $B_S(p, \rho\delta) = \emptyset$ and the lemma holds trivially. Otherwise, let $p_i \in S$ be a point such that $d(p, p_i) \leq \rho\delta$. Observe that $B_S(p, \rho\delta) \subseteq B_S(p_i, 2\rho\delta)$. Also, $|B_S(p_i, 2\rho\delta)| \leq c^{2+\lceil \log_2 \rho \rceil} |B_S(p_i, \delta/2)|$ by the

definition of the expansion constant. Because each point in C is separated by δ, the number of points in $B_S(p, \rho\delta) \cap C$ is bounded by the number of disjoint balls of radius $\delta/2$ that can be packed into $B_S(p, \rho\delta)$. In the worst case, this packing is perfect, and

$$|B_S(p, \rho\delta)| \leq \frac{|B_S(p_i, 2\rho\delta)|}{|B_S(p_i, \delta/2)|} \leq c^{2+\lceil \log_2 \rho \rceil}. \tag{4}$$

■

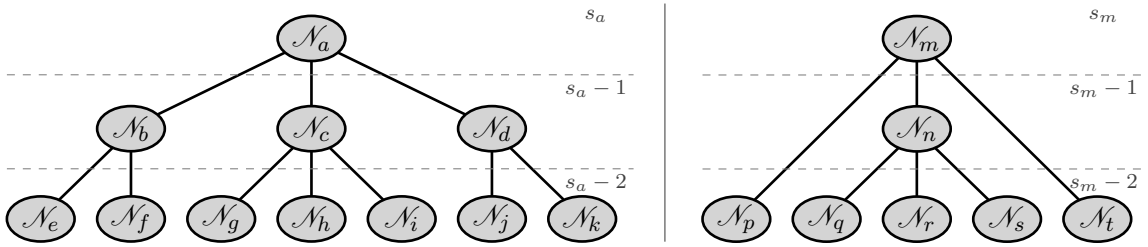
3. Tree Imbalance

It is well-known that imbalance in trees leads to degradation in performance; for instance, a *kd*-tree node with every descendant in its left child except one is effectively useless. A *kd*-tree full of nodes like this will perform abysmally for nearest neighbor search, and it is not hard to generate a pathological dataset that will cause a *kd*-tree of this sort.

This sort of imbalance applies to all types of trees, not just *kd*-trees. In our situation, we are interested in a better understanding of this imbalance for cover trees, and thus endeavor to introduce a more formal measure of imbalance which is correlated with tree performance. Numerous measures of tree imbalance have already been established; one example is that proposed by Colless (1982), and another is Sackin's index (Sackin, 1972), but we aim to capture a different measure of imbalance that uses the leveled structure of the cover tree.

We already know each node in a cover tree is indexed with an integer level (or scale). In the explicit representation of the cover tree, each non-leaf node has children at a lower level. But these children need not be strictly one level lower; see Figure 2. In Figure 2a, each cover tree node has children that are strictly one level lower; we will refer to this as a *perfectly balanced cover tree*. Figure 2b, on the other hand, contains the node $\mathscr{N}_m$ which has two children with scale two less than s_m. We will refer to this as an *imbalanced cover tree*. Note that in our definition, the balance of a cover tree has nothing to do with differing number of descendants in each child branch but instead only missing levels.

An imbalanced cover tree can happen in practice, and in the worst cases, the imbalance may be far worse than the simple graphs of Figure 2. Consider a dataset with a single



(a) Balanced cover tree.

(b) Imbalanced cover tree.

Figure 2: Balanced and imbalanced cover trees.

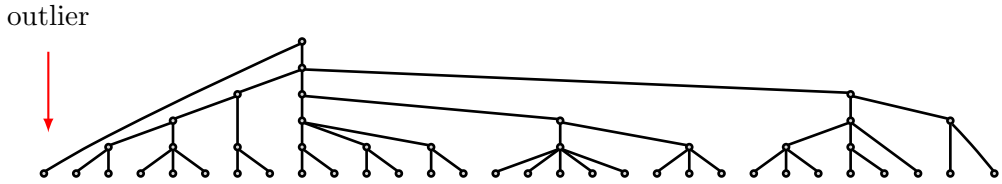


Figure 3: Single-outlier cover tree.

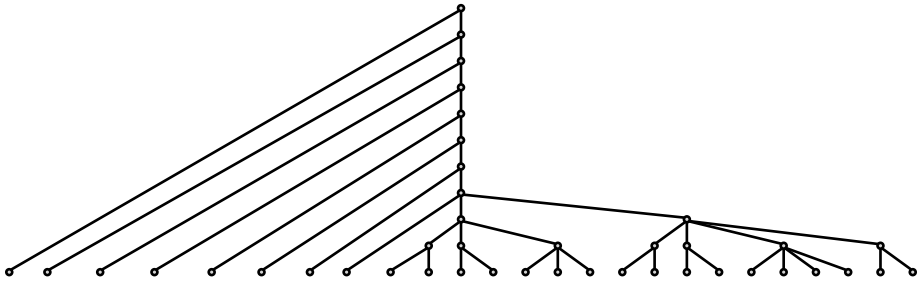


Figure 4: A multiple-outlier cover tree.

outlier which is very far away from all of the other points[3]. Figure 3 shows what happens in this situation: the root node has two children; one of these children has only the outlier as a descendant, and the other child has the rest of the points in the dataset as a descendant. In fact, it is easy to find datasets with a handful of outliers that give rise to a chain-like structure at the top of the tree: see Figure 4 for an illustration[4].

A tree that has this chain-like structure all the way down, which is similar to the *kd*-tree example at the beginning of this section, is going to perform horrendously; motivated by this observation, we define a measure of tree imbalance.

Definition 2 *The cover node imbalance $I_n(\mathscr{N}_i)$ for a cover tree node $\mathscr{N}_i$ with scale s_i in the cover tree $\mathscr{T}$ is defined as the cumulative number of missing levels between the node and its parent $\mathscr{N}_p$ (which has scale s_p). If the node is a leaf (that is, $s_i = -\infty$), then the number of missing levels is defined as the difference between s_p and $s_{\min} - 1$ where $s_{\min}$ is the smallest scale of a non-leaf node in $\mathscr{T}$. If $\mathscr{N}_i$ is the root of the tree, then the cover node imbalance is 0. Explicitly written, this calculation is*

$$I_n(\mathscr{N}_i) = \begin{cases} s_p - s_i - 1 & \text{if } \mathscr{N}_i \text{ is not a leaf and not the root node} \\ \max(s_p - s_{\min} - 1,\ 0) & \text{if } \mathscr{N}_i \text{ is a leaf} \\ 0 & \text{if } \mathscr{N}_i \text{ is the root node.} \end{cases} \tag{5}$$

3. Note also that for an outlier sufficiently far away, the expansion constant is $N - 1$, so we should expect poor performance with the cover tree anyway.
4. As a side note, this behavior is not limited to cover trees, and can happen to mean-split *kd*-trees too, especially in higher dimensions. In addition, for this scenario to arise with cover trees, it must be true that $c \sim O(N)$.

		Imbalance		
Dataset	d	$N = 5k$	$N = 50k$	$N = 500k$
lcdm	3	4.48	5.15	5.24
sdss	4	2.17	2.81	2.97
power	7	5.41	6.46	4.50
susy	18	0.74	0.76	0.86
randu	10	0.23	0.22	0.59
higgs	29	0.99	1.68	1.56
covertype	54	1.322	1.766	2.495
mnist	784	0.99	1.67	2.09

Table 2: Empirically calculated tree imbalances, normalized by N.

This simple definition of cover node imbalance is easy to calculate, and using it, we can generalize to a measure of imbalance for the full tree.

Definition 3 *The cover tree imbalance $I_t(\mathscr{T})$ for a cover tree $\mathscr{T}$ is defined as the cumulative number of missing levels in the tree. This can be expressed as a function of cover node imbalances easily:*

$$I_t(\mathscr{T}) = \sum_{\mathscr{N}_i \in \mathscr{T}} I_n(\mathscr{N}_i). \tag{6}$$

A perfectly balanced cover tree $\mathscr{T}_b$ with no missing levels has imbalance $I_t(\mathscr{T}_b) = 0$ (for instance, Figure 2a). A worst-case cover tree $\mathscr{T}_w$ which is entirely a chain-like structure with maximum scale $s_{\max}$ and minimum scale $s_{\min}$ will have imbalance $I_t(\mathscr{T}_w) \sim N(s_{\max} - s_{\min})$. Because of this chain-like structure, each level has only one node and thus there are at least N levels; or, $s_{\max} - s_{\min} \geq N$, meaning that in the worst case the imbalance is quadratic in N.[5]

However, for most real-world datasets with the cover tree implementation in **mlpack** (Curtin et al., 2013a) and the reference implementation (Beygelzimer et al., 2006), the tree imbalance is near-linear with the number of points. We have constructed cover trees on N uniformly subsampled points from a variety of datasets and calculated the imbalance; see Table 2 for the results. Ten trials were performed for each dataset and each N, and the mean imbalance is given. These results are normalized with respect to N, for which the values of 5000, 50000, and 500000 were chosen. The 'power', 'susy', 'higgs', and 'covertype' datasets are found in the UCI Machine Learning Repository (Bache and Lichman, 2013), the 'mnist' dataset is from LeCun et al. (2000), the 'lcdm' and 'sdss' datasets are Sloan Digital Sky Survey data (Adelman-McCarthy et al., 2008), and the 'randu' dataset is randomly-generated uniformly-distributed data in 10 dimensions. The imbalances on each of these datasets tend to be near-linear.

Currently, no cover tree construction algorithm specifically aims to minimize imbalance.

5. Note that in this situation, $c \sim N$ also.

Algorithm 1 The standard pruning dual-tree traversal for cover trees.

Input: query node $\mathscr{N}_q$, set of reference nodes R
Output: none

$s_r^{\max} \leftarrow \max_{\mathscr{N}_r \in R} s_r$
if $(s_q < s_r^{\max})$ **then**
  {Perform a reference recursion.}
  for each $\mathscr{N}_r \in R$ **do**
    `BaseCase(`p_q`,` p_r`)`
  end for
  $R_r \leftarrow \{\mathscr{N}_r \in R : s_r = s_r^{\max}\}$
  $R_{r-1} \leftarrow \{\mathscr{C}(\mathscr{N}_r) : \mathscr{N}_r \in R_r\} \cup (R \setminus R_r)$
  $R'_{r-1} \leftarrow \{\mathscr{N}_r \in R_{r-1} : \texttt{Score}(\mathscr{N}_q, \mathscr{N}_r) \neq \infty\}$
  recurse with $\mathscr{N}_q$ and R'_{r-1}
else
  {Perform a query recursion.}
  for each $\mathscr{N}_{qc} \in \mathscr{C}(\mathscr{N}_q)$ **do**
    $R' \leftarrow \{\mathscr{N}_r \in R : \texttt{Score}(\mathscr{N}_{qc}, \mathscr{N}_r) \neq \infty\}$
    recurse with $\mathscr{N}_{qc}$ and R'
  end for
end if

4. General Runtime Bound

Perhaps more interesting than measures of tree imbalance is the way cover trees are actually used in dual-tree algorithms. Although cover trees were originally intended for nearest neighbor search (See Algorithm `Find-All-Nearest`, Beygelzimer et al., 2006), they can be adapted to a wide variety of problems: minimum spanning tree calculation (March et al., 2010), approximate nearest neighbor search (Ram et al., 2009b), Gaussian processes posterior calculation (Moore and Russell, 2014), and max-kernel search (Curtin and Ram, 2014) are some examples. Further, through the tree-independent dual-tree algorithm abstraction of Curtin et al. (2013b), other existing dual-tree algorithms can easily be adapted for use with cover trees.

In the framework of tree-independent dual-tree algorithms, all that is necessary to describe a dual-tree algorithm is a point-to-point base case function (`BaseCase()`) and a node-to-node pruning rule (`Score()`). These functions, which are often very straightforward, are then paired with a type of tree and a pruning dual-tree traversal to produce a working algorithm. In later sections, we will consider specific examples.

When using cover trees, the typical pruning dual-tree traversal is an adapted form of the original nearest neighbor search algorithm (see `Find-All-Nearest`, Beygelzimer et al., 2006); this traversal is implemented in both the cover tree reference implementation and in the more flexible **mlpack** library (Curtin et al., 2013a). The problem-independent traversal is given in Algorithm 1 and was originally presented by Curtin and Ram (2014). Initially, it is called with the root of the query tree and a reference set R containing only the root of the reference tree.

This dual-tree recursion is a depth-first recursion in the query tree and a breadth-first recursion in the reference tree; to this end, the recursion maintains one query node $\mathscr{N}_q$ and a reference set R. The set R may contain reference nodes with many different scales; the maximum scale in the reference set is $s_r^{\max}$ (line 3). Each single recursion will descend either the query tree or the reference tree, not both; the conditional in line 4, which determines whether the query or reference tree will be recursed, is aimed at keeping the relative scales of query nodes and reference nodes close.

Keeping the query and reference scales close is both beneficial for the later theory and intuitively reasonable: recursing too quickly in the either the query or reference node will unnecessarily duplicate work. Suppose we recurse many levels down the query tree before recursing down the reference tree, giving us a set of query nodes we are considering. For *each* of these query nodes, we will then need to descend the reference tree. Because these query nodes are close together (with respect to the reference nodes we are considering, which are of larger scale and thus further apart), the pruning decisions at each level of recursion are likely to be the same for each query node. Therefore, recursing too far in the query tree may cause a large amount of duplicated work. The symmetric argument applies for recursing too far in the reference tree before recursing in the query tree. This justifies the approach of keeping the query and reference scales approximately equal.

A query recursion (lines 13–18) is straightforward: for each child $\mathscr{N}_{qc}$ of $\mathscr{N}_q$, the node combinations $(\mathscr{N}_{qc}, \mathscr{N}_r)$ are scored for each $\mathscr{N}_r$ in the reference set R. If possible, these combinations are pruned to form the set R' (line 17) by checking the output of the `Score()` function, and then the algorithm recurses with $\mathscr{N}_{qc}$ and R'.

A reference recursion (lines 4–12) is similar to a query recursion, but the pruning strategy is significantly more complicated. Given R, we calculate R_r, which is the set of nodes in R that have scale $s_r^{\max}$. We expand each node in R_r to construct R_{r-1}: this is the set of children of all nodes in R_r. This set is then combined with $R \setminus R_r$ (that is, the set of references nodes not at scale $s_r^{\max}$) to produce R_{r-1}. Each node in R_{r-1} is then scored and pruned if possible, resulting in the pruned reference set R'_{r-1}. The algorithm then recurses with $\mathscr{N}_q$ and R'_{r-1}.

The reference recursion only recurses into the top-level subset of the reference nodes in order to preserve the separation invariant. It is easy to show that every pair of points held in nodes in R is separated by at least $2^{s_r^{\max}}$:

Lemma 2 *For all distinct nodes $\mathscr{N}_i, \mathscr{N}_j \in R$ (in the context of Algorithm 1) which contain points p_i and p_j, respectively, $d(p_i, p_j) > 2^{s_r^{\max}}$, with $s_r^{\max}$ defined as in line 3.*

Proof This proof is by induction. If $|R| = 1$, such as during the first reference recursion, the result obviously holds. Now consider any reference set R and assume the statement of the lemma holds for this set R, and define $s_r^{\max}$ as the maximum scale of any node in R. Construct the set R_{r-1} as in line 10 of Algorithm 1; if $|R_{r-1}| \leq 1$, then R_{r-1} satisfies the desired property.

Otherwise, take any $\mathscr{N}_i, \mathscr{N}_j$ in R_{r-1}, with points p_i and p_j, respectively, and scales s_i and s_j, respectively. Clearly, if $s_i = s_j = s_r^{\max} - 1$, then by the separation invariant $d(p_i, p_j) > 2^{s_r^{\max}-1}$.

Now suppose that $s_i < s_r^{\max} - 1$. This implies that there exists some implicit cover tree node with point p_i and scale $s_r^{\max} - 1$ (as well as an implicit child of this node p_i with scale

$s_r^{\max} - 2$ and so forth until one of these implicit nodes has child p_i with scale s_i). Because the separation invariant applies to both implicit and explicit representations of the tree, we conclude that $d(p_i, p_j) > 2^{s_r^{\max}} - 1$. The same argument may be made for the case where $s_j < s_r^{\max} - 1$, with the same conclusion.

We may therefore conclude that each point of each node in R_{r-1} is separated by $2^{s_r^{\max}-1}$. Note that $R'_{r-1} \subseteq R_{r-1}$ and that $R \setminus R_{r-1} \subseteq R$ in order to see that this condition holds for all nodes in R'_{r-1}.

Because we have shown that the condition holds for the initial reference set and for any reference set produced by a reference recursion (which will be R at some other level of recursion), we have shown that the statement of the lemma is true.

Note that in this proof, we have considered the child reference set R_{r-1}, not the original reference set R, and shown that with respect to $s_r^{\max}$ as defined by R (not R_{r-1}), all nodes are separated by $2^{s_r^{\max}-1}$. Then, in the frame of the next recursion where $R \leftarrow R_{r-1}$, the lemma will hold, as $s_r^{\max}$ will then be the maximum scale present in R. ■

This observation means that the set of points P held by all nodes in R is always a subset of $C_{s_r^{\max}}$. This fact will be useful in our later runtime proofs.

Next, we develop notions with which to understand the behavior of the cover tree dual-tree traversal when the datasets are of significantly different scale distributions.

If the datasets are similar in scale distribution (that is, inter-point distances tend to follow the same distribution), then the recursion will alternate between query recursions and reference recursions. But if the query set contains points which are, in general, much farther apart than the reference set, then the recursion will start with many query recursions before reaching a reference recursion. The converse case also holds. We are interested in formalizing this notion of scale distribution; therefore, define the following dataset-dependent constants for the query set S_q and the reference set S_r:

- η_q: the largest pairwise distance in S_q
- δ_q: the smallest nonzero pairwise distance in S_q
- η_r: the largest pairwise distance in S_r
- δ_r: the smallest nonzero pairwise distance in S_r

These constants are directly related to the aspect ratio of the datasets; indeed, η_q/δ_q is exactly the aspect ratio of S_q. Further, let us define and bound the top and bottom levels of each tree:

- The *top scale* s_q^T of the query tree $\mathscr{T}_q$ is such that as $\lceil \log_2(\eta_q) \rceil - 1 \leq s_q^T \leq \lceil \log_2(\eta_q) \rceil$.
- The *minimum scale* of the query tree $\mathscr{T}_q$ is defined as $s_q^{\min} = \lceil \log_2(\delta_q) \rceil$.
- The top scale s_r^T of the reference tree $\mathscr{T}_r$ is such that as $\lceil \log_2(\eta_r) \rceil - 1 \leq s_r^T \leq \lceil \log_2(\eta_r) \rceil$.
- The minimum scale of the reference tree $\mathscr{T}_r$ is defined as $s_r^{\min} = \lceil \log_2(\delta_r) \rceil$.

Note that the minimum scale is not the minimum scale of *any* cover tree node (that would be $-\infty$), but the minimum scale of any non-leaf node in the tree.

Suppose that our datasets are of a similar scale distribution: $s_q^T = s_r^T$, and $s_q^{\min} = s_r^{\min}$. In this setting we will have alternating query and reference recursions. But if this is not the case, then we have extra reference recursions before the first query recursion or after the last query recursion (situations where both these cases happen are possible). Motivated by this observation, let us quantify these extra reference recursions:

Lemma 3 *For a dual-tree algorithm with $|S_q| \sim |S_r| \sim O(N)$ using cover trees and the traversal given in Algorithm 1, the number of extra reference recursions that happen before the first query recursion is bounded by*

$$\min\left(O(N), \log_2(\eta_r/\eta_q) - 1\right). \tag{7}$$

Proof The first query recursion happens once $s_q \geq s_r^{\max}$. The number of reference recursions before the first query recursion is then bounded as the number of levels in the reference tree between s_r^T and s_q^T that have at least one explicit node. Because there are $O(N)$ nodes in the reference tree, the number of levels cannot be greater than $O(N)$ and thus the result holds.

The second bound holds by applying the definitions of s_r^T and s_q^T to the expression $s_r^T - s_q^T - 1$:

$$\begin{aligned} s_r^T - s_q^T - 1 &\leq \lceil \log_2(\eta_r) \rceil - (\lceil \log_2(\eta_q) \rceil - 1) - 1 & (8) \\ &\leq \log_2(\eta_r) + 1 - \log_2(\eta_q) & (9) \end{aligned}$$

which gives the statement of the lemma after applying logarithmic identities. ■

Note that the $O(N)$ bound may be somewhat loose, but it suffices for our later purposes. Now let us consider the other case:

Lemma 4 *For a dual-tree algorithm with $|S_q| \sim |S_r| \sim O(N)$ using cover trees and the traversal given in Algorithm 1, the number of extra reference recursions that happen after the last query recursion is bounded by*

$$\max\left(\min\left(O(N \log_2(\delta_q/\delta_r)), O(N^2)\right),\ 0\right). \tag{10}$$

For convenience, we define a term that encapsulates this bound.

Definition 4 *Define θ as a bound on the number of extra reference recursions that happen after the last query recursion. Then,*

$$\theta = \max\left\{\min\left(O(N \log_2(\delta_q/\delta_r)), O(N^2)\right),\ 0\right\}. \tag{11}$$

Proof Our goal here is to count the number of reference recursions after the final query recursion at level $s_q^{\min}$; the first of these reference recursions is at scale $s_r^{\max} = s_q^{\min}$. Because query nodes are not pruned in this traversal, each reference recursion we are counting will be duplicated over the whole set of $O(N)$ query nodes. The first part of the bound follows by observing that $s_q^{\min} - s_r^{\min} \leq \lceil \log_2(\delta_q) \rceil - \lceil \log_2(\delta_r) \rceil - 1 \leq \log_2(\delta_q / \delta_r)$.

The second part follows by simply observing that there are $O(N)$ reference nodes. ■

These two previous lemmas allow us a better understanding of what happens as the reference set and query set become different. Lemma 3 shows that the number of extra recursions caused by a reference set with larger pairwise distances than the query set (η_r larger than η_q) is modest; on the other hand, Lemma 4 shows that for each extra level in the reference tree below $s_q^{\min}$, $O(N)$ extra recursions are required. Using these lemmas and this intuition, we will prove general runtime bounds for the cover tree traversal.

Theorem 1 *Given a reference set S_r of size $O(N)$ with an expansion constant c_r and a set of queries S_q of size $O(N)$, a standard cover tree based dual-tree algorithm (Algorithm 1) takes*

$$O\left(c_r^4 |R^*| \chi \psi (N + I_t(\mathscr{T}_q) + \theta)\right) \tag{12}$$

time, where $|R^*|$ is the maximum size of the reference set R (line 1) during the dual-tree recursion, χ is the maximum possible runtime of `BaseCase()`*, ψ is the maximum possible runtime of* `Score()`*, and θ is defined as in Lemma 4.*

Proof First, split the algorithm into two parts: reference recursions (lines 4–12) and query recursions (lines 13–18). The runtime of the algorithm is the runtime of a reference recursion times the total number of reference recursions plus the total runtime of all query recursions.

Consider a reference recursion (lines 4–12). Define R^* to be the largest set R for any scale $s_r^{\max}$ and any query node $\mathscr{N}_q$ during the course of the algorithm; then, it is true that $|R| \leq |R^*|$. The work done in the base case loop from lines 6–8 is thus $O(\chi |R|) \leq O(\chi |R^*|)$. Then, lines 10 and 11 take $O(c_r^4 \psi |R|) \leq O(c_r^4 \psi |R^*|)$ time, because each reference node has up to c_r^4 children. So, one full reference recursion takes $O(c_r^4 \psi \chi |R^*|)$ time.

Now, note that there are $O(N)$ nodes in $\mathscr{T}_q$. Thus, line 17 is visited $O(N)$ times. The amount of work in line 16, like in the reference recursion, is bounded as $O(c_r^4 \psi |R^*|)$. Therefore, the total runtime of all query recursions is $O(c_r^4 \psi |R^*| N)$.

Lastly, we must bound the total number of reference recursions. Reference recursions happen in three cases: *(1)* $s_r^{\max}$ is greater than the scale of the root of the query tree (no query recursions have happened yet); *(2)* $s_r^{\max}$ is less than or equal to the scale of the root of the query tree, but is greater than the minimum scale of the query tree that is not $-\infty$; *(3)* $s_r^{\max}$ is less than the minimum scale of the query tree that is not $-\infty$.

First, consider case *(1)*. Lemma 3 shows that the number of reference recursions of this type is bounded by $O(N)$. Although there is also a bound that depends on the sizes of the datasets, we only aim to show a linear runtime bound, so the $O(N)$ bound is sufficient here.

Next, consider case *(2)*. In this situation, each query recursion implies at least one reference recursion before another query recursion. For some query node $\mathscr{N}_q$, the exact number of reference recursions before the children of $\mathscr{N}_q$ are recursed into is bounded above

by $I_n(\mathscr{N}_q) + 1$: if $\mathscr{N}_q$ has imbalance 0, then it is exactly one level below its parent, and thus there is only one reference recursion. On the other hand, if $\mathscr{N}_q$ is many levels below its parent, then it is possible that a reference recursion may occur for each level in between; this is a maximum of $I_n(\mathscr{N}_q) + 1$.

Because each query node in $\mathscr{T}_q$ is recursed into once, the total number of reference recursions before each query recursion is

$$\sum_{\mathscr{N}_q \in \mathscr{T}_q} I_n(\mathscr{N}_q) + 1 = I_t(\mathscr{T}_q) + O(N) \tag{13}$$

since there are $O(N)$ nodes in the query tree.

Lastly, for case *(3)*, we may refer to Lemma 4, giving a bound of θ reference recursions in this case.

We may now combine these results for the runtime of a query recursions with the total number of reference recursions in order to give the result of the theorem:

$$O\left(c_r^4 |R^*| \psi\chi \left(N + I_t(\mathscr{T}_q) + \theta\right)\right) + O\left(c_r^4 |R^*| \psi N\right) \sim O\left(c_r^4 |R^*| \psi\chi \left(N + I_t(\mathscr{T}_q) + \theta\right)\right). \tag{14}$$

■

When we consider the monochromatic case (where $S_q = S_r$), the results trivially simplify.

Corollary 1 *Given the situation of Theorem 1 but with $S_q = S_r = S$ so that $c_q = c_r = c$ and $\mathscr{T}_q = \mathscr{T}_r = \mathscr{T}$, a dual-tree algorithm using the standard cover tree traversal (Algorithm 1) takes*

$$O\left(c^4 |R^*| \chi\psi \left(N + I_t(\mathscr{T})\right)\right) \tag{15}$$

time, where $|R^*|$ is the maximum size of the reference set R (line 1) during the dual-tree recursion, χ is the maximum possible runtime of `BaseCase()`*, and ψ is the maximum possible runtime of* `Score()`*.*

An intuitive understanding of these bounds is best achieved by first considering the monochromatic case (this case arises, for instance, in all-nearest-neighbor search). The linear dependence on N arises from the fact that all query nodes must be visited. The dependence on the reference tree, however, is encapsulated by the term $c^4|R^*|$, with $|R^*|$ being the maximum size of the reference set R; this value must be derived for each specific problem. The poor performance of trees on datasets with large c (or, in the worst case, $c \sim N$) is then captured in both of those terms. These datasets for which trees perform poorly may also have a high cover tree imbalance $I_t(\mathscr{T})$; the linear dependence of runtime on imbalance is thus sensible for datasets where trees perform well.

The bichromatic case ($S_q \neq S_r$) is a slightly more complex result which deserves a bit more attention. The intuition for all terms except θ remain virtually the same.

The term θ captures the effect of query and reference datasets with different widths, and has one unfortunate corner case: when $\delta_q > \eta_r$, then the query tree must be entirely descended before any reference recursion. This results in a bound of the form $O(N \log(\eta_r/\delta_r))$,

or $O(N^2)$ (see Lemma 4). This is because the reference tree must be descended separately for each query point.

The quantity $|R^*|$ bounds the amount of work that needs to be done for each recursion. In the worst case, $|R^*|$ can be N. However, dual-tree algorithms rely on branch-and-bound techniques to prune away work (lines 11 and 16 in Algorithm 1). A small value of $|R^*|$ will imply that the algorithm is extremely successful in pruning away work. An (upper) bound on $|R^*|$ (and the algorithm's success in pruning work) will depend on the problem and the data. As we will show, bounding $|R^*|$ is often possible. For many dual-tree algorithms, $\chi \sim \psi \sim O(1)$; often, cached sufficient statistics (Moore, 2000) can enable $O(1)$ runtime implementations of `BaseCase()` and `Score()`.

These results hold for any dual-tree algorithm regardless of the problem. Hence, the runtime of any dual-tree algorithm can be bounded no more tightly than $O(N)$ with our bound, which matches the intuition that answering $O(N)$ queries will take at least $O(N)$ time. For a particular problem and data, if c_r, $|R^*|$, χ, and ψ are bounded by constants independent of N and θ is no more than linear in N (for large enough N), then the dual-tree algorithm for that problem has a runtime linear in N. Our theoretical result separates out the problem-dependent and the problem-independent elements of the runtime bound, which allows us to simply plug in the problem-dependent bounds in order to get runtime bounds for any dual-tree algorithm without requiring an analysis from scratch.

Our results are similar to that of Ram et al. (2009a), but those results depend on a quantity called the *constant of bichromaticity*, denoted κ, which has unclear relation to cover tree imbalance. The dependence on κ is given as $c_q^{4\kappa}$, which is not a good bound, especially because κ may be much greater than 1 in the bichromatic case (where $S_q \neq S_r$).

The more recent results of Curtin and Ram (2014) are more related to these results, but they depend on the *inverse constant of bichromaticity* ν which suffers from the same problem as κ. Although the dependence on ν is linear (that is, $O(\nu N)$), bounding ν is difficult and it is not true that $\nu = 1$ in the monochromatic case.

The quantity ν corresponds to the maximum number of reference recursions between a single query recursion, and κ corresponds to the maximum number of query recursions between a single reference recursion. The respective proofs that use these constants then apply them as a worst-case measure for the whole algorithm: when using κ, Ram et al. (2009a) assume that *every* reference recursion may be followed by κ query recursions; similarly, Curtin and Ram (2014) assume that *every* query recursion may be followed by ν reference recursions. Here, we have simply used $I_t(\mathscr{T}_q)$ and θ as an exact summation of the total extra reference recursions, which gives us a much tighter bound than ν or κ on the running time of the whole algorithm.

Further, both ν and κ are difficult to empirically calculate and require an entire run of the dual-tree algorithm. On the other hand, bounding $I_t(\mathscr{T}_q)$ (and θ) can be done in one pass of the tree (assuming the tree is already built). Thus, not only is our bound tighter when the cover tree imbalance is sublinear in N, it more closely reflects the actual behavior of dual-tree algorithms, and the constants which it depends upon are straightforward to calculate.

In the following sections, we will apply our results to specific problems and show the utility of our bound in simplifying runtime proofs for dual-tree algorithms.

Algorithm 2 Nearest neighbor search `BaseCase()`

Input: query point p_q, reference point p_r, list of candidate neighbors N and distances D

Output: distance d between p_q and p_r

if $d(p_q, p_r) < D[p_q]$ **and** `BaseCase(`p_q`,` p_r`)` not yet called **then**
  $D[p_q] \leftarrow d(p_q, p_r)$, and $N[p_q] \leftarrow p_r$
end if
return $d(p_q, p_r)$

Algorithm 3 Nearest neighbor search `Score()`

Input: query node $\mathscr{N}_q$, reference node $\mathscr{N}_r$
Output: a score for the node combination $(\mathscr{N}_q, \mathscr{N}_r)$, or ∞ if the combination should be pruned

if $d_{\min}(\mathscr{N}_q, \mathscr{N}_r) < B(\mathscr{N}_q)$ **then**
  return $d_{\min}(\mathscr{N}_q, \mathscr{N}_r)$
end if
return ∞

5. Nearest Neighbor Search

The standard task of nearest neighbor search can be simply described: given a query set S_q and a reference set S_r, for each query point $p_q \in S_q$, find the nearest neighbor p_r in the reference set S_r. The task is well-studied and well-known, and there exist numerous approaches for both exact and approximate nearest neighbor search, including the cover tree nearest neighbor search algorithm due to Beygelzimer et al. (2006). We will consider that algorithm, but in a tree-independent sense as given by Curtin et al. (2013b); this means that to describe the algorithm, we require only a `BaseCase()` and `Score()` function; these are given in Algorithms 2 and 3, respectively. The point-to-point `BaseCase()` function compares a query point p_q and a reference point p_r, updating the list of candidate neighbors for p_q if necessary.

The node-to-node `Score()` function determines if the entire subtree of nodes under the reference node $\mathscr{N}_r$ can improve the candidate neighbors for all descendant points of the query node $\mathscr{N}_q$; if not, the node combination is pruned. The `Score()` function depends on the function $d_{\min}(\cdot, \cdot)$, which represents the minimum possible distance between any two descendants of two nodes. Its definition for cover tree nodes is

$$d_{\min}(\mathscr{N}_q, \mathscr{N}_r) = d(p_q, p_r) - 2^{s_q+1} - 2^{s_r+1}. \tag{16}$$

Given a type of tree and traversal, these two functions store the current nearest neighbor candidates in the array N and their distances in the array D. (See Curtin et al., 2013b, for a more complete discussion of how this algorithm works and a proof of correctness.) The `Score()` function depends on a bound function $B(\mathscr{N}_q)$ which represents the maximum distance that could possibly improve a nearest neighbor candidate for any descendant point

of the query node $\mathscr{N}_q$. The standard bound function $B(\mathscr{N}_q)$ used for cover trees is adapted from Beygelzimer et al. (2006):

$$B(\mathscr{N}_q) := D[p_q] + 2^{s_q+1} \tag{17}$$

In this formulation, the query node $\mathscr{N}_q$ holds the the query point p_q, the quantity $D[p_q]$ is the current nearest neighbor candidate distance for the query point p_q, and 2^{s_q+1} corresponds to the furthest descendant distance of $\mathscr{N}_q$. For notational convenience in the following proof, take $c_{qr} = \max((\max_{p_q \in S_q} c'_r), c_r)$, where c'_r is the expansion constant of the set $S_r \cup \{p_q\}$.

Theorem 2 *Using cover trees, the standard cover tree pruning dual-tree traversal, and the nearest neighbor search* `BaseCase()` *and* `Score()` *as given in Algorithms 2 and 3, respectively, and also given a reference set S_r of size $O(N)$ with expansion constant c_r, and a query set S_q of size $O(N)$, the running time of the algorithm is bounded by $O(c_r^4 c_{qr}^5 (N + I_t(\mathscr{T}_q) + \theta))$ with $I_t(\mathscr{T}_q)$ and θ defined as in Definition 3 and Lemma 4, respectively.*

Proof The running time of `BaseCase()` and `Score()` are clearly $O(1)$. Due to Theorem 1, we therefore know that the runtime of the algorithm is bounded by $O(c_r^4 |R^*| (N + I_t(\mathscr{T}_q) + \theta))$. Thus, the only thing that remains is to bound the maximum size of the reference set, $|R^*|$.

Assume that when R^* is encountered, the maximum reference scale is $s_r^{\max}$ and the query node is $\mathscr{N}_q$. Every node $\mathscr{N}_r \in R^*$ satisfies the property enforced in line 11 that $d_{\min}(\mathscr{N}_q, \mathscr{N}_r) \leq B(\mathscr{N}_q)$. Using the definition of $d_{\min}(\cdot, \cdot)$ and $B(\cdot)$, we expand the equation. Note that p_q is the point held in $\mathscr{N}_q$ and p_r is the point held in $\mathscr{N}_r$. Also, take $\hat{p}_r$ to be the current nearest neighbor candidate for p_q; that is, $D[p_q] = d(p_q, \hat{p}_r)$ and $N[p_q] = \hat{p}_r$. Then,

$$\begin{aligned} d_{\min}(\mathscr{N}_q, \mathscr{N}_r) &\leq B(\mathscr{N}_q) && (18) \\ d(p_q, p_r) &\leq d(p_q, \hat{p}_r) + 2^{s_q+1} + 2^{s_r+1} + 2^{s_q+1} && (19) \\ &\leq d(p_q, \hat{p}_r) + 2(2^{s_r^{\max}+1}) && (20) \end{aligned}$$

where the last step follows because $s_q + 1 \leq s_r^{\max}$ and $s_r \leq s_r^{\max}$. Define the set of points P as the points held in each node in R^* (that is, $P = \{p_r \in \mathscr{P}(\mathscr{N}_r) : \mathscr{N}_r \in R^*\}$). Then, we can write

$$P \subseteq B_{S_r}(p_q, d(p_q, \hat{p}_r) + 2(2^{s_r^{\max}+1})). \tag{21}$$

Suppose that the true nearest neighbor is p_r^* and $d(p_q, p_r^*) > 2^{s_r^{\max}+1}$. Then, p_r^* must be held as a descendant point of some node in R^* which holds some point $\tilde{p}_r$. Using the triangle inequality,

$$d(p_q, \hat{p}_r) \leq d(p_q, \tilde{p}_r) \leq d(p_q, p_r^*) + d(\tilde{p}_r, p_r^*) \leq d(p_q, p_r^*) + 2^{s_r^{\max}+1}. \tag{22}$$

This gives that $P \subseteq B_{S_r \cup \{p_q\}}(p_q, d(p_q, p_r^*) + 3(2^{s_r^{\max}+1}))$. The previous step is necessary: to apply the definition of the expansion constant, the ball must be centered at a point in the set; now, the center (p_q) is part of the set.

$$|B_{S_r \cup \{p_q\}}(p_q, d(p_q, p_r^*) + 3(2^{s_r^{\max}+1}))| \leq |B_{S_r \cup \{p_q\}}(p_q, 4d(p_q, p_r^*))| \quad (23)$$
$$\leq c_{qr}^3 |B_{S_r \cup \{p_q\}}(p_q, d(p_q, p_r^*)/2)| \quad (24)$$

which follows because the expansion constant of the set $S_r \cup \{p_q\}$ is bounded above by c_{qr}. Next, we know that p_r^* is the closest point to p_q in $S_r \cup \{p_q\}$; thus, there cannot exist a point $p_r' \neq p_q \in S_r \cup \{p_q\}$ such that $p_r' \in B_{S_{qr}}(p_q, d(p_q, p_r^*)/2)$ because that would imply that $d(p_q, p_r') < d(p_q, p_r^*)$, which is a contradiction. Thus, the only point in the ball is p_q, and we have that $|B_{S_r \cup \{p_q\}}(p_q, d(p_q, p_r^*)/2)| = 1$, giving the result that $|R| \leq c_{qr}^3$ in this case.

The other case is when $d(p_q, p_r^*) \leq 2^{s_r^{\max}+1}$, which means that $d(p_q, \hat{p}_r) \leq 2^{s_r^{\max}+2}$. Note that $P \in C_{s_r^{\max}}$, and therefore

$$P \subseteq B_{S_r}(p_q, d(p_q, p_r^*) + 3(2^{s_r^{\max}+1})) \cap C_{s_r^{\max}} \quad (25)$$
$$\subseteq B_{S_r}(p_q, 4(2^{s_r^{\max}+1})) \cap C_{s_r^{\max}}. \quad (26)$$

Every point in $C_{s_r^{\max}}$ is separated by at least $2^{s_r^{\max}}$. Using Lemma 1 with $\delta = 2^{s_r^{\max}}$ and $\rho = 8$ yields that $|P| \leq c_r^5$. This gives the result, because $c_r^5 \leq c_{qr}^5$. ■

In the monochromatic case where $S_q = S_r$[6], the bound is $O(c^9(N + I_t(\mathscr{T}))$ because $c = c_r = c_{qr}$ and $\theta = 0$. For well-behaved trees where $I_t(\mathscr{T}_q)$ is linear or sublinear in N, this represents the current tightest worst-case runtime bound for nearest neighbor search.

6. Approximate Kernel Density Estimation

Ram et al. (2009a) present a clever technique for bounding the running time of approximate kernel density estimation based on the properties of the kernel, when the kernel is shift-invariant and satisfies a few assumptions. We will restate these assumptions and provide an adapted proof using Theorem 1, which gives a tighter bound.

Approximate kernel density estimation is a common application of dual-tree algorithms (Gray and Moore, 2003, 2001). Given a query set S_q, a reference set S_r of size N, and a kernel function $\mathcal{K}(\cdot, \cdot)$, the true kernel density estimate for a query point p_q is given as

$$f^*(p_q) = \sum_{p_r \in S_r} \mathcal{K}(p_q, p_r). \quad (27)$$

In the case of an infinite-tailed kernel $\mathcal{K}(\cdot, \cdot)$, the exact computation cannot be accelerated; thus, attention has turned towards tractable approximation schemes. Two simple schemes for the approximation of $f^*(p_q)$ are well-known: *absolute value approximation* and *relative value approximation*. Absolute value approximation requires that each density estimate $f(p_q)$ is within ϵ of the true estimate $f^*(p_q)$:

$$|f(p_q) - f^*(p_q)| < \epsilon \ \ \forall p_q \in S_q. \quad (28)$$

6. In the monochromatic case, we do not take a point as its own nearest neighbor, so slight modification of `BaseCase()` is necessary. The runtime bound result remains unchanged.

Relative value approximation is a more flexible approximation scheme; given some parameter ϵ, the requirement is that each density estimate is within a relative tolerance of $f^*(p_q)$:

$$\frac{|f(p_q) - f^*(p_q)|}{|f^*(p_q)|} < \epsilon \ \ \forall p_q \in S_q. \tag{29}$$

Kernel density estimation is related to the well-studied problem of kernel summation, which can also be solved with dual-tree algorithms (Lee and Gray, 2006, 2009). In both of those problems, regardless of the approximation scheme, simple geometric observations can be made to accelerate computation: when $\mathcal{K}(\cdot, \cdot)$ is shift-invariant, faraway points have very small kernel evaluations. Thus, trees can be built on S_q and S_r, and node combinations can be pruned when the nodes are far apart while still obeying the error bounds.

In the following two subsections, we will separately consider both the absolute value approximation scheme and the relative value approximation scheme, under the assumption of a shift-invariant kernel $\mathcal{K}(p_q, p_r) = \mathcal{K}(\|p_q - p_r\|)$ which is monotonically decreasing and non-negative. In addition, we assume that there exists some bandwidth h such that $\mathcal{K}(d)$ must be concave for $d \in [0, h]$ and convex for $d \in [h, \infty)$. This assumption implies that the magnitude of the derivative $|\mathcal{K}'(d)|$ is maximized at $d = h$. These are not restrictive assumptions; most standard kernels fall into this class, including the Gaussian, exponential, and Epanechnikov kernels.

6.1 Absolute Value Approximation

A tree-independent algorithm for solving approximate kernel density estimation with absolute value approximation under the previous assumptions on the kernel is given as a `BaseCase()` function in Algorithm 4 and a `Score()` function in Algorithm 5 (a correctness proof can be found in Curtin et al., 2013b). The list f_p holds partial kernel density estimates for each query point, and the list f_n holds partial kernel density estimates for each query node. At the beginning of the dual-tree traversal, the lists f_p and f_n, which are both of size $O(N)$, are each initialized to 0. As the traversal proceeds, node combinations are pruned if the difference between the maximum kernel value $\mathcal{K}(d_{\min}(\mathcal{N}_q, \mathcal{N}_r))$ and the minimum kernel value $\mathcal{K}(d_{\max}(\mathcal{N}_q, \mathcal{N}_r))$ is sufficiently small (line 3). If the node combination can be pruned, then the partial node estimate is updated (line 4). When node combinations cannot be pruned, `BaseCase()` may be called, which simply updates the partial point estimate with the exact kernel evaluation (line 3).

After the dual-tree traversal, the actual kernel density estimates f must be extracted. This can be done by traversing the query tree and calculating $f(p_q) = f_p(p_q) + \sum_{\mathcal{N}_i \in T} f_n(\mathcal{N}_i)$, where T is the set of nodes in $\mathcal{T}_q$ that have p_q as a descendant. Each query node needs to be visited only once to perform this calculation; it may therefore be accomplished in $O(N)$ time.

Note that this version is far simpler than other dual-tree algorithms that have been proposed for approximate kernel density estimation (see, for instance, Gray and Moore, 2003); however, this version is sufficient for our runtime analysis. Real-world implementations, such as the one found in **mlpack** (Curtin et al., 2013a), tend to be far more complex.

Algorithm 4 Approximate kernel density estimation `BaseCase()`

Input: query point p_q, reference point p_r, list of kernel point estimates $\hat{f}_p$
Output: kernel value $\mathcal{K}(p_q, p_r)$
$f_p(p_q) \leftarrow f_p(p_q) + \mathcal{K}(p_q, p_r)$
return $\mathcal{K}(p_q, p_r)$

Algorithm 5 Absolute-value approximate kernel density estimation `Score()`

Input: query node $\mathscr{N}_q$, reference node $\mathscr{N}_r$, list of node kernel estimates $\hat{f}_n$
Output: a score for the node combination $(\mathscr{N}_q, \mathscr{N}_r)$, or ∞ if the combination should be pruned
if $\mathcal{K}(d_{\min}(\mathscr{N}_q, \mathscr{N}_r)) - \mathcal{K}(d_{\max}(\mathscr{N}_q, \mathscr{N}_r)) < \epsilon$ **then**
  $f_n(\mathscr{N}_q) \leftarrow f_n(\mathscr{N}_q) + |\mathscr{D}^p(\mathscr{N}_r)| \left(\mathcal{K}(d_{\min}(\mathscr{N}_q, \mathscr{N}_r)) + \mathcal{K}(d_{\max}(\mathscr{N}_q, \mathscr{N}_r))\right) / 2$
  return ∞
end if
return $\mathcal{K}(d_{\min}(\mathscr{N}_q, \mathscr{N}_r)) - \mathcal{K}(d_{\max}(\mathscr{N}_q, \mathscr{N}_r))$

Theorem 3 *Assume that $\mathcal{K}(\cdot, \cdot)$ is a kernel with bandwidth h satisfying the assumptions of the previous subsection. Then, given a query set S_q of size $O(N)$ and a reference set S_r of size $O(N)$ with expansion constant c_r, and using the approximate kernel density estimation `BaseCase()` and `Score()` as given in Algorithms 4 and 5, respectively, with the traversal given in Algorithm 1, the running time of approximate kernel density estimation for some error parameter ϵ is bounded by $O(c_r^{8+\lceil \log_2 \zeta \rceil}(N + I_t(\mathscr{T}_q) + \theta))$ with $\zeta = -\mathcal{K}'(h)\mathcal{K}^{-1}(\epsilon)\epsilon^{-1}$, $I_t(\mathscr{T}_q)$ defined as in Definition 3, and θ defined as in Lemma 4.*

Proof It is clear that `BaseCase()` and `Score()` both take $O(1)$ time, so Theorem 1 implies the total runtime of the dual-tree algorithm is bounded by $O(c_r^4 |R^*|(N + I_t(\mathscr{T}_q) + \theta))$. Thus, we will bound $|R^*|$ using techniques related to those used by Ram et al. (2009a). The bounding of $|R^*|$ is split into two sections: first, we show that when the scale $s_r^{\max}$ is small enough, R^* is empty. Second, we bound R^* when $s_r^{\max}$ is larger.

The `Score()` function is such that any node in R^* for a given query node $\mathscr{N}_q$ obeys

$$\mathcal{K}(d_{\min}(\mathscr{N}_q, \mathscr{N}_r)) - \mathcal{K}(d_{\max}(\mathscr{N}_q, \mathscr{N}_r)) \geq \epsilon. \tag{30}$$

Thus, we are interested in the maximum possible value $\mathcal{K}(a) - \mathcal{K}(b)$ for a fixed value of $b - a > 0$. Due to our assumptions, the maximum value of $\mathcal{K}'(\cdot)$ is $\mathcal{K}'(h)$; therefore, the maximum possible value of $\mathcal{K}(a) - \mathcal{K}(b)$ is when the interval $[a, b]$ is centered on h. This allows us to say that $\mathcal{K}(a) - \mathcal{K}(b) \leq \epsilon$ when $(b - a) \leq (-\epsilon / \mathcal{K}'(h))$. Note that

$$\begin{aligned} d_{\max}(\mathscr{N}_q, \mathscr{N}_r) - d_{\min}(\mathscr{N}_q, \mathscr{N}_r) &\leq d(p_q, p_r) + 2^{s_r^{\max}+1} - d(p_q, p_r) + 2^{s_r^{\max}+1} && (31) \\ &\leq 2^{s_r^{\max}+2}. && (32) \end{aligned}$$

Therefore, $R^* = \emptyset$ when $2^{s_r^{\max}+2} \leq -\epsilon / \mathcal{K}'(h)$, or when $s_r^{\max} \leq \log_2(-\epsilon / \mathcal{K}'(h)) - 2$. Consider, then, the case when $s_r^{\max} > \log_2(-\epsilon / \mathcal{K}'(h)) - 2$. Because of the pruning rule,

for any $\mathscr{N}_r \in R^*$, $\mathcal{K}(d_{\min}(\mathscr{N}_q, \mathscr{N}_r)) > \epsilon$; we may refactor this by applying definitions to show $d(p_q, p_r) < \mathcal{K}^{-1}(\epsilon) + 2^{s_r^{\max}+1}$. Therefore, bounding the number of points in the set $B_{S_r}(p_q, \mathcal{K}^{-1}(\epsilon) + 2^{s_r^{\max}+1}) \cap C_{s_r^{\max}}$ is sufficient to bound $|R^*|$. For notational convenience, define $\omega = (\mathcal{K}^{-1}(\epsilon)/2^{s_r^{\max}+1}) + 1$, and the statement may be more concisely written as $B_{S_r}(p_q, \omega 2^{s_r^{\max}+1}) \cap C_{s_r^{\max}}$.

Using Lemma 1 with $\delta = 2^{s_r^{\max}}$ and $\rho = 2\omega$ gives $|R^*| = c_r^{3+\lceil \log_2 \omega \rceil}$.

The value ω is maximized when $s_r^{\max}$ is minimized. Using the lower bound on $s_r^{\max}$, ω is bounded as $\omega = -2\mathcal{K}'(h)\mathcal{K}^{-1}(\epsilon)\epsilon^{-1}$. Finally, with $\zeta = -\mathcal{K}'(h)\mathcal{K}^{-1}(\epsilon)\epsilon^{-1}$, we are able to conclude that $|R^*| \leq c_r^{3+\lceil \log_2 (2\zeta) \rceil} = c_r^{4+\lceil \log_2 \zeta \rceil}$. Therefore, the entire dual-tree traversal takes $O(c_r^{8+\lceil \log_2 \zeta \rceil}(N + \theta))$ time.

The postprocessing step to extract the estimates $f(\cdot)$ requires one traversal of the tree $\mathscr{T}_r$; the tree has $O(N)$ nodes, so this takes only $O(N)$ time. This is less than the runtime of the dual-tree traversal, so the runtime of the dual-tree traversal dominates the algorithm's runtime, and the theorem holds. ■

The dependence on ϵ (through ζ) is expected: as $\epsilon \to 0$ and the search becomes exact, ζ diverges both because ϵ^{-1} diverges and also because $\mathcal{K}^{-1}(\epsilon)$ diverges, and the runtime goes to the worst-case $O(N^2)$; exact kernel density estimation means no nodes can be pruned at all.

For the Gaussian kernel with bandwidth σ defined by $\mathcal{K}_g(d) = \exp(-d^2/(2\sigma^2))$, ζ does not depend on the kernel bandwidth; only the approximation parameter ϵ. For this kernel, $h = \sigma$ and therefore $-\mathcal{K}_g'(h) = \sigma^{-1}e^{-1/2}$. Additionally, $\mathcal{K}_g^{-1}(\epsilon) = \sigma\sqrt{2\ln(1/\epsilon)}$. This means that for the Gaussian kernel, $\zeta = \sqrt{(-2\ln\epsilon)/(e\epsilon^2)}$. Again, as $\epsilon \to 0$, the runtime diverges; however, note that there is no dependence on the kernel bandwidth σ. To demonstrate the relationship of runtime to ϵ, see that for a reasonably chosen $\epsilon = 0.05$, the runtime is approximately $O(c_r^{8.89}(N+\theta))$; for $\epsilon = 0.01$, the runtime is approximately $O(c_r^{11.52}(N+\theta))$. For very small $\epsilon = 0.00001$, the runtime is approximately $O(c_r^{22.15}(N+\theta))$.

Next, consider the exponential kernel: $\mathcal{K}_l(d) = \exp(-d/\sigma)$. For this kernel, $h = 0$ (that is, the kernel is always convex), so then $\mathcal{K}_l'(h) = \sigma^{-1}$. Simple algebraic manipulation gives $\mathcal{K}_l^{-1}(\epsilon) = -\sigma\ln\epsilon$, resulting in $\zeta = -\mathcal{K}_l'(h)\mathcal{K}_l^{-1}(\epsilon)\epsilon^{-1} = \epsilon^{-1}\ln\epsilon$. So both the exponential and Gaussian kernels do not exhibit dependence on the bandwidth.

To understand the lack of dependence on kernel bandwidth more intuitively, consider that as the kernel bandwidth increases, two things happen: *(a)* the reference set R becomes empty at larger scales, and *(b)* $\mathcal{K}^{-1}(\epsilon)$ grows, allowing less pruning at higher levels. These effects are opposite, and for the Gaussian and exponential kernels they cancel each other out, giving the same bound regardless of bandwidth.

6.2 Relative Value Approximation

Approximate kernel density estimation using relative-value approximation may be bounded by reducing the absolute-value approximation algorithm (in linear time or less) to relative-value approximation. This is the same strategy as performed by Ram et al. (2009a).

Algorithm 6 Relative-value approximate kernel density estimation `Score()`

1: **Input:** query node $\mathscr{N}_q$, reference node $\mathscr{N}_r$, list of node kernel estimates $\hat{f}_n$
2: **Output:** a score for the node combination $(\mathscr{N}_q, \mathscr{N}_r)$, or ∞ if the combination should be pruned
3: **if** $\mathcal{K}(d_{\min}(\mathscr{N}_q, \mathscr{N}_r)) - \mathcal{K}(d_{\max}(\mathscr{N}_q, \mathscr{N}_r)) < \epsilon \mathcal{K}^{\max}$ **then**
4: $\quad f_n(\mathscr{N}_q) \leftarrow f_n(\mathscr{N}_q) + |\mathscr{D}^p(\mathscr{N}_r)| \left(\mathcal{K}(d_{\min}(\mathscr{N}_q, \mathscr{N}_r)) + \mathcal{K}(d_{\max}(\mathscr{N}_q, \mathscr{N}_r))\right) / 2$
5: $\quad$ **return** ∞
6: **end if**
7: **return** $\mathcal{K}(d_{\min}(\mathscr{N}_q, \mathscr{N}_r)) - \mathcal{K}(d_{\max}(\mathscr{N}_q, \mathscr{N}_r))$

First, we must establish a `Score()` function for relative value approximation. The difference between Equations 28 and 29 is the division by the term $|f^*(p_q)|$. But we can quickly bound $|f^*(p_q)|$:

$$|f^*(p_q)| \geq N\mathcal{K}\left(\max_{p_r \in S_r} d(p_q, p_r)\right). \tag{33}$$

This is clearly true: each point in S_r must contribute more than $\mathcal{K}(\max_{p_r \in S_r} d(p_q, p_r))$ to $f^*(p_q)$. Now, we may revise the relative approximation condition in Equation 29:

$$|f(p_q) - f^*(p_q)| \leq \epsilon \mathcal{K}^{\max} \tag{34}$$

where $\mathcal{K}^{\max}$ is lower bounded by $\mathcal{K}(\max_{p_r \in S_r} d(p_q, p_r))$. Assuming we have some estimate $\mathcal{K}^{\max}$, this allows us to create a `Score()` algorithm, given in Algorithm 6.

Using this, we may prove linear runtime bounds for relative value approximate kernel density estimation.

Theorem 4 *Assume that $\mathcal{K}(\cdot, \cdot)$ is a kernel satisfying the same assumptions as Theorem 3. Then, given a query set S_q and a reference set S_r both of size $O(N)$, it is possible to perform relative value approximate kernel density estimation (satisfying the condition of Equation 29) in $O(N)$ time, assuming that the expansion constant c_r of S_r is not dependent on N.*

Proof It is easy to see that Theorem 3 may be adapted to the very slightly different `Score()` rule of Algorithm 6 while still providing an $O(N)$ bound. With that `Score()` function, the dual-tree algorithm will return relative-value approximate kernel density estimates satisfying Equation 29.

We now turn to the calculation of $\mathcal{K}^{\max}$. Given the cover trees $\mathscr{T}_q$ and $\mathscr{T}_r$ with root nodes $\mathscr{N}_r^R$ and $\mathscr{N}_r^R$, respectively, we may calculate a suitable $\mathcal{K}^{\max}$ value in constant time:

$$\mathcal{K}^{\max} = d_{\max}(\mathscr{N}_q^R, \mathscr{N}_r^R) = d(p_q^R, p_r^R) + 2^{s_q^{\max}+1} + 2^{s_r^{\max}+1}. \tag{35}$$

This proves the statement of the theorem. ■

In this case, we have not shown tighter bounds because the algorithm we have proposed is not useful in practice. For an example of a better relative-value approximate kernel density estimation dual-tree algorithm, see the work of Gray and Moore (2003).

Algorithm 7 Range search `BaseCase()`

Input: query point p_q, reference point p_r, range sets $N[p_q]$ and range $[l, u]$
Output: distance d between p_q and p_r
if $d(p_q, p_r) \in [r_{\min}, r_{\max}]$ **and** `BaseCase(`p_q`,` p_r`)` not yet called **then**
  $S[p_q] \leftarrow S[p_q] \cup \{p_r\}$
end if
return d

Algorithm 8 Range search `Score()`

Input: query node $\mathscr{N}_q$, reference node $\mathscr{N}_r$
Output: a score for the node combination $(\mathscr{N}_q, \mathscr{N}_r)$, or ∞ if the combination should be pruned
if $d_{\min}(\mathscr{N}_q, \mathscr{N}_r) \in [l, u]$ or $d_{\max}(\mathscr{N}_q, \mathscr{N}_r) \in [l, u]$ **then**
  return $d_{\min}(\mathscr{N}_q, \mathscr{N}_r)$
end if
return ∞

7. Range Search and Range Count

In the range search problem, the task is to find the set of reference points

$$S[p_q] = \{p_r \in S_r : d(p_q, p_r) \in [l, u]\} \tag{36}$$

for each query point p_q, where $[l, u]$ is the given range. The range count problem is practically identical, but only the size of the set, $|S[p_q]|$, is desired. Our proof works for both of these algorithms similarly, but we will focus on range search. A `BaseCase()` and `Score()` function are given in Algorithms 7 and 8, respectively (a correctness proof can be found in Curtin et al., 2013b). The sets $N[p_q]$ (for each p_q) are initialized to $\emptyset$ at the beginning of the traversal.

In order to bound the running time of dual-tree range search, we require better notions for understanding the difficulty of the problem. Observe that if the range is sufficiently large, then for every query point p_q, $S[p_q] = S_r$. Clearly, for $S_q \sim S_r \sim O(N)$, this cannot be solved in anything less than quadratic time simply due to the time required to fill each output array $S[p_q]$. Define the maximum result size for a given query set S_q, reference set S_r, and range $[l, u]$ as

$$|S_{\max}| = \max_{p_q \in S_q} |S[p_q]|. \tag{37}$$

Small $|S_{\max}|$ implies an easy problem; large $|S_{\max}|$ implies a difficult problem. For bounding the running time of range search, we require one more notion of difficulty, related to how $|S_{\max}|$ changes due to changes in the range $[l, u]$.

Definition 5 *For a range search problem with query set S_q, reference set S_r, range $[l, u]$, and results $S[p_q]$ for each query point p_q given as*

$$S[p_q] = \{p_r : p_r \in S_r, l \le d(p_q, p_r) \le u\}, \tag{38}$$

define the α-expansion of the range set $S[p_q]$ as the slightly larger set

$$S^{\alpha}[p_q] = \{p_r : p_r \in S_r, (1-\alpha)l \leq d(p_q, p_r) \leq (1+\alpha)u\}. \tag{39}$$

When the α-expansion of the set $S_{\max}$ is approximately the same size as $S_{\max}$, then the problem would not be significantly more difficult if the range $[l, u]$ was increased slightly. Using these notions, then, we may now bound the running time of range search.

Theorem 5 *Given a reference set S_r of size $O(N)$ with expansion constant c_r, and a query set S_q of size $O(N)$, a search range of $[l, u]$, and using the range search* `BaseCase()` *and* `Score()` *as given in Algorithms 7 and 8, respectively, with the standard cover tree pruning dual-tree traversal as given in Algorithm 1, and also assuming that for some $\alpha > 0$,*

$$|S^{\alpha}[p_q] \setminus S[p_q]| \leq C \ \ \forall \, p_q \in S_q, \tag{40}$$

the running time of range search or range count is bounded by

$$O\left(c_r^4 \max\left(c_r^{4+\beta}, |S_{\max}| + C\right)(N + I_t(\mathscr{N}_q) + \theta)\right) \tag{41}$$

with θ defined as in Lemma 4, $\beta = \lceil \log_2(1 + \alpha^{-1}) \rceil$, and $S_{\max}$ as defined in Equation 37.

Proof Both `BaseCase()` (Algorithm 7) and `Score()` (Algorithm 8) take $O(1)$ time. Therefore, using Lemma 1, we know that the runtime of the algorithm is bounded by $O(c_r^4 |R^*|(N + I_t(\mathscr{N}_q) + \theta))$. As with the previous proofs, then, our only task is to bound the maximum size of the reference set, $|R^*|$.

By the pruning rule, for a query node $\mathscr{N}_q$, the reference set R^* is made up of reference nodes $\mathscr{N}_r$ that are within a margin of $2^{s_q+1} + 2^{s_r+1} \leq 2^{s_r^{\max}+2}$ of the range $[l, u]$. Given that p_r is the point in $\mathscr{N}_r$,

$$p_r \in \left(B_{S_r}(p_q, u + 2^{s_r^{\max}+2}) \cap C_{s_r^{\max}}\right) \setminus \left(B_{S_r}(p_q, l - 2^{s_r^{\max}+2}) \cap C_{s_r^{\max}}\right). \tag{42}$$

A bound on the number of elements in this set is a bound on $|R^*|$. First, consider the case where $u \leq \alpha^{-1} 2^{s_r^{\max}+2}$. Ignoring the smaller ball, take $\delta = 2^{s_r^{\max}}$ and $\rho = 4(1 + \alpha^{-1})$ and apply Lemma 1 to produce the bound

$$|R^*| \leq c_r^{4+\lceil \log_2(1+\alpha^{-1}) \rceil}. \tag{43}$$

Now, consider the other case: $u > \alpha^{-1} 2^{s_r^{\max}+1}$. This means

$$B_{S_r}(p_q, u + 2^{s_r^{\max}+1}) \setminus B_{S_r}(p_q, l - 2^{s_r^{\max}+1}) \subseteq B_{S_r}(p_q, (1+\alpha)u) \setminus B_{S_r}(p_q, (1-\alpha)l). \tag{44}$$

This set is necessarily a subset of $S^{\alpha}[p_q]$; by assumption, the number of points in this set is bounded above by $|S_{\max}| + C$. We may then conclude that $|R^*| \leq |S_{\max}| + C$. By taking the maximum of the sizes of $|R^*|$ in both cases above, we obtain the statement of the theorem. ■

This bound displays both the expected dependence on c_r and $|S_{\max}|$. As the largest range set $S_{\max}$ increases in size (with the worst case being $S_{\max} \sim N$), the runtime degenerates to quadratic. But for adequately small $S_{\max}$ the runtime is instead dependent on c_r and the parameter C of the α-expansion of $S_{\max}$. This situation leads to a simplification.

Corollary 2 *For sufficiently small $|S_{\max}|$ and sufficiently small C, the runtime of range search under the conditions of Theorem 5 simplifies to*

$$O(c_r^{8+\beta}(N + I_t(\mathscr{N}_q) + \theta)). \tag{45}$$

In this setting we can more easily consider the relation of the running time to α. Consider $\alpha = (1/3)$; this yields a running time of $O(c^8(N+\theta))$. $\alpha = (1/7)$ yields $O(c^9(N+I_t(\mathscr{N}_q+\theta))$, $\alpha = (1/15)$ yields $O(c^{10}(N + I_t(\mathscr{N}_q) + \theta))$, and so forth. As α gets smaller, the exponent on c gets larger, and diverges as $\alpha \to 0$.

For reasonable runtime it is necessary that the α-expansion of $S_{\max}$ be bounded. This is because the dual-tree recursion must retain reference nodes which may contain descendants in the range set $S[p_q]$ for some query p_q. The parameter C of the α-expansion allows us to bound the number of reference nodes of this type, and if α increases but C remains small enough that Corollary 2 applies, then we are able to obtain tighter running bounds.

8. Conclusion

We have presented a unified framework for bounding the runtimes of dual-tree algorithms that use cover trees and the standard cover tree pruning dual-tree traversal (Algorithm 1). In order to produce an understandable bound, we have introduced the notion of cover tree imbalance; one possible interesting direction of future work is to empirically and theoretically minimize this quantity by way of modified tree construction algorithms; this is likely to provide both tighter runtime bounds and also accelerated empirical results.

Our main result, Theorem 1, allows plug-and-play runtime bounding of these algorithms. We have shown that Theorem 1 is useful for bounding the runtime of nearest neighbor search (Theorem 2), approximate kernel density estimation (Theorem 3), exact range count, and exact range search (Theorem 5). With our contribution, bounding a cover tree dual-tree algorithm is streamlined and only involves bounding the maximum size of the reference set, $|R^*|$.

Acknowledgements

The authors gratefully acknowledge the helpful and insightful comments of the anonymous reviewers.

References

J.K. Adelman-McCarthy, M.A. Agüeros, S.S. Allam, C.A. Prieto, K.S.J. Anderson, S.F. Anderson, J. Annis, N.A. Bahcall, C.A.L. Bailer-Jones, I.K. Baldry, et al. The sixth data

release of the Sloan Digital Sky Survey. *The Astrophysical Journal Supplement Series*, 175(2):297, 2008.

S. Amizadeh, B. Thiesson, and M. Hauskrecht. Variational dual-tree framework for large-scale transition matrix approximation. In *Proceedings of the Twenty-Eighth Annual Conference on Uncertainty in Artificial Intelligence (UAI-12)*, pages 64–73, Catalina Island, 2012.

K. Bache and M. Lichman. UCI Machine Learning Repository, 2013. `http://archive.ics.uci.edu/ml`.

A. Beygelzimer, S.M. Kakade, and J. Langford. Cover trees for nearest neighbor. In *Proceedings of the 23rd International Conference on Machine Learning (ICML '06)*, pages 97–104, Pittsburgh, 2006.

D.H. Colless. Review of 'Phylogenetics: The Theory and Practice of Phylogenetic Systematics', by E.O. Wiley. *Systematic Zoology*, 31:100–104, 1982.

R.R. Curtin and P. Ram. Dual-tree fast exact max-kernel search. *Statistical Analysis and Data Mining*, 7(4):229–253, 2014.

R.R. Curtin, J.R. Cline, N.P. Slagle, W.B. March, P. Ram, N.A. Mehta, and A.G. Gray. MLPACK: A scalable C++ machine learning library. *Journal of Machine Learning Research*, 14:801–805, 2013a.

R.R. Curtin, W.B. March, P. Ram, D.V. Anderson, A.G. Gray, and C.L. Isbell Jr. Tree-independent dual-tree algorithms. In *Proceedings of The 30th International Conference on Machine Learning (ICML '13)*, pages 1435–1443, Atlanta, 2013b.

R.R. Curtin, P. Ram, and A.G. Gray. Fast exact max-kernel search. In *Proceedings of the 13th SIAM International Conference on Data Mining (SDM '13)*, pages 1–9, Philadelphia, 2013c.

R.A. Finkel and J.L. Bentley. Quad trees a data structure for retrieval on composite keys. *Acta Informatica*, 4(1):1–9, 1974.

A.G. Gray and A.W. Moore. N-body problems in statistical learning. In *Advances in Neural Information Processing Systems 13 (NIPS 2000)*, pages 521–527, Vancouver, 2001.

A.G. Gray and A.W. Moore. Nonparametric density estimation: Toward computational tractability. In *Proceedings of the 3rd SIAM International Conference on Data Mining (SDM '03)*, pages 203–211, San Francisco, 2003.

D.R. Karger and M. Ruhl. Finding nearest neighbors in growth-restricted metrics. In *Proceedings of the Thirty-Fourth Annual ACM Symposium on Theory of Computing (STOC 2002)*, pages 741–750, Montréal, 2002.

M. Klaas, M. Briers, N. De Freitas, A. Doucet, S. Maskell, and D. Lang. Fast particle smoothing: if I had a million particles. In *Proceedings of the 23rd International Conference on Machine Learning (ICML '06)*, pages 25–29, Pittsburgh, 2006.

R. Krauthgamer and J.R. Lee. Navigating nets: simple algorithms for proximity search. In *Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA04)*, pages 798–807, New Orleans, 2004.

Y. LeCun, C. Cortes, and C.J.C. Burges. MNist dataset, 2000. `http://yann.lecun.com/exdb/mnist/`.

D. Lee and A.G. Gray. Faster Gaussian summation: Theory and Experiment. In *Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence (UAI-06)*, pages 281–288, Arlington, 2006.

D. Lee and A.G. Gray. Fast high-dimensional kernel summations using the monte carlo multipole method. *Advances in Neural Information Processing Systems 21 (NIPS 2008)*, pages 929–936, 2009.

W.B. March. *Multi-tree algorithms for computational statistics and physics.* PhD thesis, Georgia Institute of Technology, 2013.

W.B. March, P. Ram, and A.G. Gray. Fast Euclidean minimum spanning tree: algorithm, analysis, and applications. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '10)*, pages 603–612, Washington, D.C., 2010.

W.B. March, A.J. Connolly, and A.G. Gray. Fast algorithms for comprehensive n-point correlation estimates. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '12)*, pages 1478–1486, Beijing, 2012.

A.W. Moore. The Anchors hierarchy: Using the triangle inequality to survive high dimensional data. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence (UAI-00)*, pages 397–405, Stanford, 2000.

D.A. Moore and S.J. Russell. Fast Gaussian process posteriors with product trees. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence (UAI-14)*, Quebec City, July 2014.

P. Ram, D. Lee, W.B. March, and A.G. Gray. Linear-time algorithms for pairwise statistical problems. *Advances in Neural Information Processing Systems 22 (NIPS 2009)*, pages 1527–1535, 2009a.

P. Ram, D. Lee, H. Ouyang, and A.G. Gray. Rank-approximate nearest neighbor search: Retaining meaning and speed in high dimensions. In *Advances in Neural Information Processing Systems 22 (NIPS 2009)*, pages 1536–1544, Vancouver, 2009b.

M.J. Sackin. "Good" and "bad" phenograms. *Systematic Biology*, 21(2):225–226, 1972.

L. Van Der Maaten. Accelerating t-SNE using tree-based algorithms. *The Journal of Machine Learning Research*, 15(1):3221–3245, 2014.

M. Vladymyrov and M.A. Carreira-Perpinán. Linear-time training of nonlinear low-dimensional embeddings. In *Proceedings of The Seventeenth International Conference on Artificial Intelligence and Statistics, JMLR W&CP (AISTATS 2014)*, volume 33, pages 968–977, 2014.

P. Wang, D. Lee, A.G. Gray, and J.M. Rehg. Fast mean shift with accurate and stable convergence. In *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics (AISTATS 2007)*, pages 604–611, San Juan, 2007.

Journal of Machine Learning Research 16 (2015) 3299-3340 Submitted 1/15; Revised 7/15; Published 12/15

Divide and Conquer Kernel Ridge Regression: A Distributed Algorithm with Minimax Optimal Rates

Yuchen Zhang YUCZHANG@BERKELEY.EDU
Department of Electrical Engineering and Computer Science
University of California, Berkeley, Berkeley, CA 94720, USA

John Duchi JDUCHI@STANFORD.EDU
Departments of Statistics and Electrical Engineering
Stanford University, Stanford, CA 94305, USA

Martin Wainwright WAINWRIG@BERKELEY.EDU
Departments of Statistics and Electrical Engineering and Computer Science
University of California, Berkeley, Berkeley, CA 94720, USA

Editor: Hui Zou

Abstract

We study a decomposition-based scalable approach to kernel ridge regression, and show that it achieves minimax optimal convergence rates under relatively mild conditions. The method is simple to describe: it randomly partitions a dataset of size N into m subsets of equal size, computes an independent kernel ridge regression estimator for each subset using a careful choice of the regularization parameter, then averages the local solutions into a global predictor. This partitioning leads to a substantial reduction in computation time versus the standard approach of performing kernel ridge regression on all N samples. Our two main theorems establish that despite the computational speed-up, statistical optimality is retained: as long as m is not too large, the partition-based estimator achieves the statistical minimax rate over all estimators using the set of N samples. As concrete examples, our theory guarantees that the number of subsets m may grow nearly linearly for finite-rank or Gaussian kernels and polynomially in N for Sobolev spaces, which in turn allows for substantial reductions in computational cost. We conclude with experiments on both simulated data and a music-prediction task that complement our theoretical results, exhibiting the computational and statistical benefits of our approach.

Keywords: kernel ridge regression, divide and conquer, computation complexity

1. Introduction

In non-parametric regression, the statistician receives N samples of the form $\{(x_i, y_i)\}_{i=1}^N$, where each $x_i \in \mathcal{X}$ is a covariate and $y_i \in \mathbb{R}$ is a real-valued response, and the samples are drawn i.i.d. from some unknown joint distribution $\mathbb{P}$ over $\mathcal{X} \times \mathbb{R}$. The goal is to estimate a function $\widehat{f} : \mathcal{X} \to \mathbb{R}$ that can be used to predict future responses based on observing only the covariates. Frequently, the quality of an estimate $\widehat{f}$ is measured in terms of the mean-squared prediction error $\mathbb{E}[(\widehat{f}(X) - Y)^2]$, in which case the conditional expectation $f^*(x) = \mathbb{E}[Y \mid X = x]$ is optimal. The problem of non-parametric regression is a classical one, and a researchers have studied a wide range of estimators (see, for example, the books of Gyorfi et al. (2002), Wasserman (2006), or van de Geer (2000)). One class of

methods, known as regularized M-estimators (van de Geer, 2000), are based on minimizing the combination of a data-dependent loss function with a regularization term. The focus of this paper is a popular M-estimator that combines the least-squares loss with a squared Hilbert norm penalty for regularization. When working in a reproducing kernel Hilbert space (RKHS), the resulting method is known as *kernel ridge regression*, and is widely used in practice (Hastie et al., 2001; Shawe-Taylor and Cristianini, 2004). Past work has established bounds on the estimation error for RKHS-based methods (Koltchinskii, 2006; Mendelson, 2002a; van de Geer, 2000; Zhang, 2005), which have been refined and extended in more recent work (e.g., Steinwart et al., 2009).

Although the statistical aspects of kernel ridge regression (KRR) are well-understood, the computation of the KRR estimate can be challenging for large datasets. In a standard implementation (Saunders et al., 1998), the kernel matrix must be inverted, which requires $\mathcal{O}(N^3)$ time and $\mathcal{O}(N^2)$ memory. Such scalings are prohibitive when the sample size N is large. As a consequence, approximations have been designed to avoid the expense of finding an exact minimizer. One family of approaches is based on low-rank approximation of the kernel matrix; examples include kernel PCA (Schölkopf et al., 1998), the incomplete Cholesky decomposition (Fine and Scheinberg, 2002), or Nyström sampling (Williams and Seeger, 2001). These methods reduce the time complexity to $\mathcal{O}(dN^2)$ or $\mathcal{O}(d^2N)$, where $d \ll N$ is the preserved rank. The associated prediction error has only been studied very recently. Concurrent work by Bach (2013) establishes conditions on the maintained rank that still guarantee optimal convergence rates; see the discussion in Section 7 for more detail. A second line of research has considered early-stopping of iterative optimization algorithms for KRR, including gradient descent (Yao et al., 2007; Raskutti et al., 2011) and conjugate gradient methods (Blanchard and Krämer, 2010), where early-stopping provides regularization against over-fitting and improves run-time. If the algorithm stops after t iterations, the aggregate time complexity is $\mathcal{O}(tN^2)$.

In this work, we study a different decomposition-based approach. The algorithm is appealing in its simplicity: we partition the dataset of size N randomly into m equal sized subsets, and we compute the kernel ridge regression estimate $\widehat{f}_i$ for each of the $i = 1, \ldots, m$ subsets independently, with a *careful choice* of the regularization parameter. The estimates are then averaged via $\bar{f} = (1/m) \sum_{i=1}^{m} \widehat{f}_i$. Our main theoretical result gives conditions under which the average $\bar{f}$ achieves the minimax rate of convergence over the underlying Hilbert space. Even using naive implementations of KRR, this decomposition gives time and memory complexity scaling as $\mathcal{O}(N^3/m^2)$ and $\mathcal{O}(N^2/m^2)$, respectively. Moreover, our approach dovetails naturally with parallel and distributed computation: we are guaranteed superlinear speedup with m parallel processors (though we must still communicate the function estimates from each processor). Divide-and-conquer approaches have been studied by several authors, including McDonald et al. (2010) for perceptron-based algorithms, Kleiner et al. (2012) in distributed versions of the bootstrap, and Zhang et al. (2013) for parametric smooth convex optimization problems. This paper demonstrates the potential benefits of divide-and-conquer approaches for nonparametric and infinite-dimensional regression problems.

One difficulty in solving each of the sub-problems independently is how to choose the regularization parameter. Due to the infinite-dimensional nature of non-parametric problems, the choice of regularization parameter must be made with care (e.g., Hastie et al.,

2001). An interesting consequence of our theoretical analysis is in demonstrating that, even though each partitioned sub-problem is based only on the fraction N/m of samples, it is nonetheless *essential to regularize the partitioned sub-problems as though they had all N samples*. Consequently, from a local point of view, each sub-problem is under-regularized. This "under-regularization" allows the bias of each local estimate to be very small, but it causes a detrimental blow-up in the variance. However, as we prove, the m-fold averaging underlying the method reduces variance enough that the resulting estimator $\bar{f}$ still attains optimal convergence rate.

The remainder of this paper is organized as follows. We begin in Section 2 by providing background on the kernel ridge regression estimate and discussing the assumptions that underlie our analysis. In Section 3, we present our main theorems on the mean-squared error between the averaged estimate $\bar{f}$ and the optimal regression function f^*. We provide both a result when the regression function f^* belongs to the Hilbert space $\mathcal{H}$ associated with the kernel, as well as a more general oracle inequality that holds for a general f^*. We then provide several corollaries that exhibit concrete consequences of the results, including convergence rates of r/N for kernels with finite rank r, and convergence rates of $N^{-2\nu/(2\nu+1)}$ for estimation of functionals in a Sobolev space with ν-degrees of smoothness. As we discuss, both of these estimation rates are minimax-optimal and hence unimprovable. We devote Sections 4 and 5 to the proofs of our results, deferring more technical aspects of the analysis to appendices. Lastly, we present simulation results in Section 6.1 to further explore our theoretical results, while Section 6.2 contains experiments with a reasonably large music prediction experiment.

2. Background and Problem Formulation

We begin with the background and notation required for a precise statement of our problem.

2.1 Reproducing Kernels

The method of kernel ridge regression is based on the idea of a reproducing kernel Hilbert space. We provide only a very brief coverage of the basics here, referring the reader to one of the many books on the topic (Wahba, 1990; Shawe-Taylor and Cristianini, 2004; Berlinet and Thomas-Agnan, 2004; Gu, 2002) for further details. Any symmetric and positive semidefinite kernel function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ defines a reproducing kernel Hilbert space (RKHS for short). For a given distribution $\mathbb{P}$ on $\mathcal{X}$, the Hilbert space is strictly contained in $L^2(\mathbb{P})$. For each $x \in \mathcal{X}$, the function $z \mapsto K(z, x)$ is contained with the Hilbert space $\mathcal{H}$; moreover, the Hilbert space is endowed with an inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ such that $K(\cdot, x)$ acts as the representer of evaluation, meaning

$$\langle f, K(x, \cdot) \rangle_{\mathcal{H}} = f(x) \quad \text{for } f \in \mathcal{H}. \tag{1}$$

We let $\|g\|_{\mathcal{H}} := \sqrt{\langle g, g \rangle_{\mathcal{H}}}$ denote the norm in $\mathcal{H}$, and similarly $\|g\|_2 := (\int_{\mathcal{X}} g(x)^2 d\mathbb{P}(x))^{1/2}$ denotes the norm in $L^2(\mathbb{P})$. Under suitable regularity conditions, Mercer's theorem guarantees that the kernel has an eigen-expansion of the form

$$K(x, x') = \sum_{j=1}^{\infty} \mu_j \phi_j(x) \phi_j(x'),$$

where $\mu_1 \geq \mu_2 \geq \cdots \geq 0$ are a non-negative sequence of eigenvalues, and $\{\phi_j\}_{j=1}^{\infty}$ is an orthonormal basis for $L^2(\mathbb{P})$.

From the reproducing relation (1), we have $\langle \phi_j, \phi_j \rangle_{\mathcal{H}} = 1/\mu_j$ for any j and $\langle \phi_j, \phi_{j'} \rangle_{\mathcal{H}} = 0$ for any $j \neq j'$. For any $f \in \mathcal{H}$, by defining the basis coefficients $\theta_j = \langle f, \phi_j \rangle_{L^2(\mathbb{P})}$ for $j = 1, 2, \ldots$, we can expand the function in terms of these coefficients as $f = \sum_{j=1}^{\infty} \theta_j \phi_j$, and simple calculations show that

$$\|f\|_2^2 = \int_{\mathcal{X}} f^2(x) d\mathbb{P}(x) = \sum_{j=1}^{\infty} \theta_j^2, \quad \text{and} \quad \|f\|_{\mathcal{H}}^2 = \langle f, f \rangle_{\mathcal{H}} = \sum_{j=1}^{\infty} \frac{\theta_j^2}{\mu_j}.$$

Consequently, we see that the RKHS can be viewed as an elliptical subset of the sequence space $\ell^2(\mathbb{N})$ as defined by the non-negative eigenvalues $\{\mu_j\}_{j=1}^{\infty}$.

2.2 Kernel Ridge Regression

Suppose that we are given a data set $\{(x_i, y_i)\}_{i=1}^N$ consisting of N i.i.d. samples drawn from an unknown distribution $\mathbb{P}$ over $\mathcal{X} \times \mathbb{R}$, and our goal is to estimate the function that minimizes the mean-squared error $\mathbb{E}[(f(X) - Y)^2]$, where the expectation is taken jointly over (X, Y) pairs. It is well-known that the optimal function is the conditional mean $f^*(x) := \mathbb{E}[Y \mid X = x]$. In order to estimate the unknown function f^*, we consider an M-estimator that is based on minimizing a combination of the least-squares loss defined over the dataset with a weighted penalty based on the squared Hilbert norm,

$$\widehat{f} := \operatorname*{argmin}_{f \in \mathcal{H}} \left\{ \frac{1}{N} \sum_{i=1}^{N} (f(x_i) - y_i)^2 + \lambda \|f\|_{\mathcal{H}}^2 \right\}, \tag{2}$$

where $\lambda > 0$ is a regularization parameter. When $\mathcal{H}$ is a reproducing kernel Hilbert space, then the estimator (2) is known as the *kernel ridge regression estimate*, or KRR for short. It is a natural generalization of the ordinary ridge regression estimate (Hoerl and Kennard, 1970) to the non-parametric setting.

By the representer theorem for reproducing kernel Hilbert spaces (Wahba, 1990), any solution to the KRR program (2) must belong to the linear span of the kernel functions $\{K(\cdot, x_i),\ i = 1, \ldots, N\}$. This fact allows the computation of the KRR estimate to be reduced to an N-dimensional quadratic program, involving the N^2 entries of the kernel matrix $\{K(x_i, x_j),\ i, j = 1, \ldots, n\}$. On the statistical side, a line of past work (van de Geer, 2000; Zhang, 2005; Caponnetto and De Vito, 2007; Steinwart et al., 2009; Hsu et al., 2012) has provided bounds on the estimation error of $\widehat{f}$ as a function of N and λ.

3. Main Results and Their Consequences

We now turn to the description of our algorithm, followed by the statements of our main results, namely Theorems 1 and 2. Each theorem provides an upper bound on the mean-squared prediction error for any trace class kernel. The second theorem is of "oracle type," meaning that it applies even when the true regression function f^* does not belong to the Hilbert space $\mathcal{H}$, and hence involves a combination of approximation and estimation error terms. The first theorem requires that $f^* \in \mathcal{H}$, and provides somewhat sharper bounds on

the estimation error in this case. Both of these theorems apply to any trace class kernel, but as we illustrate, they provide concrete results when applied to specific classes of kernels. Indeed, as a corollary, we establish that our distributed KRR algorithm achieves minimax-optimal rates for three different kernel classes, namely finite-rank, Gaussian, and Sobolev.

3.1 Algorithm and Assumptions

The divide-and-conquer algorithm Fast-KRR is easy to describe. Rather than solving the kernel ridge regression problem (2) on all N samples, the Fast-KRR method executes the following three steps:

1. Divide the set of samples $\{(x_1, y_1), \ldots, (x_N, y_N)\}$ evenly and uniformly at random into the m disjoint subsets $S_1, \ldots, S_m \subset \mathcal{X} \times \mathbb{R}$, such that every subset contains N/m samples.

2. For each $i = 1, 2, \ldots, m$, compute the *local KRR estimate*

$$\widehat{f}_i := \operatorname*{argmin}_{f \in \mathcal{H}} \left\{ \frac{1}{|S_i|} \sum_{(x,y) \in S_i} (f(x) - y)^2 + \lambda \|f\|_{\mathcal{H}}^2 \right\}. \tag{3}$$

3. Average together the local estimates and output $\bar{f} = \frac{1}{m} \sum_{i=1}^m \widehat{f}_i$.

This description actually provides a family of estimators, one for each choice of the regularization parameter $\lambda > 0$. Our main result applies to any choice of λ, while our corollaries for specific kernel classes optimize λ as a function of the kernel.

We now describe our main assumptions. Our first assumption, for which we have two variants, deals with the tail behavior of the basis functions $\{\phi_j\}_{j=1}^{\infty}$.

Assumption A *For some $k \geq 2$, there is a constant $\rho < \infty$ such that $\mathbb{E}[\phi_j(X)^{2k}] \leq \rho^{2k}$ for all $j \in \mathbb{N}$.*

In certain cases, we show that sharper error guarantees can be obtained by enforcing a stronger condition of uniform boundedness.

Assumption A$'$ *There is a constant $\rho < \infty$ such that $\sup_{x \in \mathcal{X}} |\phi_j(x)| \leq \rho$ for all $j \in \mathbb{N}$.*

Assumption A$'$ holds, for example, when the input x is drawn from a closed interval and the kernel is translation invariant, i.e. $K(x, x') = \psi(x - x')$ for some even function ψ. Given input space $\mathcal{X}$ and kernel K, the assumption is verifiable without the data.

Recalling that $f^*(x) := \mathbb{E}[Y \mid X = x]$, our second assumption involves the deviations of the zero-mean noise variables $Y - f^*(x)$. In the simplest case, when $f^* \in \mathcal{H}$, we require only a bounded variance condition:

Assumption B *The function $f^* \in \mathcal{H}$, and for $x \in \mathcal{X}$, we have $\mathbb{E}[(Y - f^*(x))^2 \mid x] \leq \sigma^2$.*

When the function $f^* \notin \mathcal{H}$, we require a slightly stronger variant of this assumption. For each $\lambda \geq 0$, define

$$f_\lambda^* = \operatorname*{argmin}_{f \in \mathcal{H}} \left\{ \mathbb{E}\left[(f(X) - Y)^2\right] + \lambda \|f\|_{\mathcal{H}}^2 \right\}. \tag{4}$$

Note that $f^* = f_0^*$ corresponds to the usual regression function. As $f^* \in L^2(\mathbb{P})$, for each $\lambda \geq 0$, the associated mean-squared error $\sigma_\lambda^2(x) := \mathbb{E}[(Y - f_\lambda^*(x))^2 \mid x]$ is finite for almost every x. In this more general setting, the following assumption replaces Assumption B:

Assumption B′ *For any $\lambda \geq 0$, there exists a constant $\tau_\lambda < \infty$ such that $\tau_\lambda^4 = \mathbb{E}[\sigma_\lambda^4(X)]$.*

3.2 Statement of Main Results

With these assumptions in place, we are now ready for the statements of our main results. All of our results give bounds on the mean-squared estimation error $\mathbb{E}[\|\bar{f} - f^*\|_2^2]$ associated with the averaged estimate $\bar{f}$ based on an assigning $n = N/m$ samples to each of m machines. Both theorem statements involve the following three kernel-related quantities:

$$\operatorname{tr}(K) := \sum_{j=1}^{\infty} \mu_j, \quad \gamma(\lambda) := \sum_{j=1}^{\infty} \frac{1}{1 + \lambda/\mu_j}, \quad \text{and} \quad \beta_d = \sum_{j=d+1}^{\infty} \mu_j. \tag{5}$$

The first quantity is the kernel trace, which serves a crude estimate of the "size" of the kernel operator, and assumed to be finite. The second quantity $\gamma(\lambda)$, familiar from previous work on kernel regression (Zhang, 2005), is the *effective dimensionality* of the kernel K with respect to $L^2(\mathbb{P})$. Finally, the quantity β_d is parameterized by a positive integer d that we may choose in applying the bounds, and it describes the tail decay of the eigenvalues of K. For $d = 0$, note that $\beta_0 = \operatorname{tr} K$. Finally, both theorems involve a quantity that depends on the number of moments k in Assumption A:

$$b(n, d, k) := \max\left\{\sqrt{\max\{k, \log(d)\}}, \frac{\max\{k, \log(d)\}}{n^{1/2-1/k}}\right\}. \tag{6}$$

Here the integer $d \in \mathbb{N}$ is a free parameter that may be optimized to obtain the sharpest possible upper bound. (The algorithm's execution is independent of d.)

Theorem 1 *With $f^* \in \mathcal{H}$ and under Assumptions A and B, the mean-squared error of the averaged estimate $\bar{f}$ is upper bounded as*

$$\mathbb{E}\left[\left\|\bar{f} - f^*\right\|_2^2\right] \leq \left(8 + \frac{12}{m}\right)\lambda \left\|f^*\right\|_{\mathcal{H}}^2 + \frac{12\sigma^2\gamma(\lambda)}{N} + \inf_{d \in \mathbb{N}}\left\{T_1(d) + T_2(d) + T_3(d)\right\}, \tag{7}$$

where

$$T_1(d) = \frac{8\rho^4 \left\|f^*\right\|_{\mathcal{H}}^2 \operatorname{tr}(K)\beta_d}{\lambda}, \quad T_2(d) = \frac{4\left\|f^*\right\|_{\mathcal{H}}^2 + 2\sigma^2/\lambda}{m}\left(\mu_{d+1} + \frac{12\rho^4 \operatorname{tr}(K)\beta_d}{\lambda}\right), \quad \textit{and}$$

$$T_3(d) = \left(Cb(n, d, k)\frac{\rho^2\gamma(\lambda)}{\sqrt{n}}\right)^k \mu_0 \left\|f^*\right\|_{\mathcal{H}}^2 \left(1 + \frac{2\sigma^2}{m\lambda} + \frac{4\left\|f^*\right\|_{\mathcal{H}}^2}{m}\right),$$

and C denotes a universal (numerical) constant.

Theorem 1 is a general result that applies to any trace-class kernel. Although the statement appears somewhat complicated at first sight, it yields concrete and interpretable guarantees on the error when specialized to particular kernels, as we illustrate in Section 3.3.

Before doing so, let us make a few heuristic arguments in order to provide intuition. In typical settings, the term $T_3(d)$ goes to zero quickly: if the number of moments k is suitably large and number of partitions m is small—say enough to guarantee that $(b(n,d,k)\gamma(\lambda)/\sqrt{n})^k = \mathcal{O}(1/N)$—it will be of lower order. As for the remaining terms, at a high level, we show that an appropriate choice of the free parameter d leaves the first two terms in the upper bound (7) dominant. Note that the terms μ_{d+1} and β_d are decreasing in d while the term $b(n,d,k)$ increases with d. However, the increasing term $b(n,d,k)$ grows only logarithmically in d, which allows us to choose a fairly large value without a significant penalty. As we show in our corollaries, for many kernels of interest, as long as the number of machines m is not "too large," this tradeoff is such that $T_1(d)$ and $T_2(d)$ are also of lower order compared to the two first terms in the bound (7). In such settings, Theorem 1 guarantees an upper bound of the form

$$\mathbb{E}\left[\left\|\bar{f} - f^*\right\|_2^2\right] = \mathcal{O}(1) \cdot \Big[\underbrace{\lambda \left\|f^*\right\|_{\mathcal{H}}^2}_{\text{Squared bias}} + \underbrace{\frac{\sigma^2\gamma(\lambda)}{N}}_{\text{Variance}} \Big]. \tag{8}$$

This inequality reveals the usual bias-variance trade-off in non-parametric regression; choosing a smaller value of $\lambda > 0$ reduces the first squared bias term, but increases the second variance term. Consequently, the setting of λ that minimizes the sum of these two terms is defined by the relationship

$$\lambda \left\|f^*\right\|_{\mathcal{H}}^2 \;\simeq\; \sigma^2 \frac{\gamma(\lambda)}{N}. \tag{9}$$

This type of fixed point equation is familiar from work on oracle inequalities and local complexity measures in empirical process theory (Bartlett et al., 2005; Koltchinskii, 2006; van de Geer, 2000; Zhang, 2005), and when λ is chosen so that the fixed point equation (9) holds this (typically) yields minimax optimal convergence rates (Bartlett et al., 2005; Koltchinskii, 2006; Zhang, 2005; Caponnetto and De Vito, 2007). In Section 3.3, we provide detailed examples in which the choice λ^* specified by equation (9), followed by application of Theorem 1, yields minimax-optimal prediction error (for the Fast-KRR algorithm) for many kernel classes.

We now turn to an error bound that applies without requiring that $f^* \in \mathcal{H}$. In order to do so, we introduce an auxiliary variable $\bar{\lambda} \in [0, \lambda]$ for use in our analysis (the algorithm's execution does not depend on $\bar{\lambda}$, and in our ensuing bounds we may choose any $\bar{\lambda} \in [0,\lambda]$ to give the sharpest possible results). Let the radius $R = \left\|f^*_{\bar{\lambda}}\right\|_{\mathcal{H}}$, where the population (regularized) regression function $f^*_{\bar{\lambda}}$ was previously defined (4). The theorem requires a few additional conditions to those in Theorem 1, involving the quantities $\mathrm{tr}(K)$, $\gamma(\lambda)$ and β_d defined in Eq. (5), as well as the error moment $\tau_{\bar{\lambda}}$ from Assumption B′. We assume that the triplet (m,d,k) of positive integers satisfy the conditions

$$\begin{aligned} &\beta_d \leq \frac{\lambda}{(R^2 + \tau^2_{\bar{\lambda}}/\lambda)N}, \quad \mu_{d+1} \leq \frac{1}{(R^2 + \tau^2_{\bar{\lambda}}/\lambda)N}, \\ &m \leq \min\left\{ \frac{\sqrt{N}}{\rho^2\gamma(\lambda)\log(d)}, \frac{N^{1-\frac{2}{k}}}{(R^2 + \tau^2_{\bar{\lambda}}/\lambda)^{2/k}(b(n,d,k)\rho^2\gamma(\lambda))^2} \right\}. \end{aligned} \tag{10}$$

We then have the following result:

Theorem 2 *Under condition* (10)*, Assumption A with $k \geq 4$, and Assumption B′, for any $\bar{\lambda} \in [0, \lambda]$ and $q > 0$ we have*

$$\mathbb{E}\left[\left\|\bar{f} - f^*\right\|_2^2\right] \leq \left(1 + \frac{1}{q}\right) \inf_{\|f\|_{\mathcal{H}} \leq R} \|f - f^*\|_2^2 + (1+q)\, \mathcal{E}_{N,m}(\lambda, \bar{\lambda}, R, \rho) \tag{11}$$

where the residual term is given by

$$\mathcal{E}_{N,m}(\lambda, \bar{\lambda}, R, \rho) := \left(\left(4 + \frac{C}{m}\right)(\lambda - \bar{\lambda})R^2 + \frac{C\gamma(\lambda)\rho^2\tau_{\bar{\lambda}}^2}{N} + \frac{C}{N}\right), \tag{12}$$

and C denotes a universal (numerical) constant.

Remarks: Theorem 2 is an oracle inequality, as it upper bounds the mean-squared error in terms of the error $\inf_{\|f\|_{\mathcal{H}} \leq R} \|f - f^*\|_2^2$, which may only be obtained by an oracle knowing the sampling distribution $\mathbb{P}$, along with the residual error term (12).

In some situations, it may be difficult to verify Assumption B′. In such scenarios, an alternative condition suffices. For instance, if there exists a constant $\kappa < \infty$ such that $\mathbb{E}[Y^4] \leq \kappa^4$, then under condition (10), the bound (11) holds with $\tau_{\bar{\lambda}}^2$ replaced by $\sqrt{8\,\mathrm{tr}(K)^2 R^4 \rho^4 + 8\kappa^4}$—that is, with the alternative residual error

$$\widetilde{\mathcal{E}}_{N,m}(\lambda, \bar{\lambda}, R, \rho) := \left(\left(2 + \frac{C}{m}\right)(\lambda - \bar{\lambda})R^2 + \frac{C\gamma(\lambda)\rho^2\sqrt{8\,\mathrm{tr}(K)^2 R^4 \rho^4 + 8\kappa^4}}{N} + \frac{C}{N}\right). \tag{13}$$

In essence, if the response variable Y has sufficiently many moments, the prediction mean-square error $\tau_{\bar{\lambda}}^2$ in the statement of Theorem 2 can be replaced by constants related to the size of $\left\|f_{\bar{\lambda}}^*\right\|_{\mathcal{H}}$. See Section 5.2 for a proof of inequality (13).

In comparison with Theorem 1, Theorem 2 provides somewhat looser bounds. It is, however, instructive to consider a few special cases. For the first, we may assume that $f^* \in \mathcal{H}$, in which case $\|f^*\|_{\mathcal{H}} < \infty$. In this setting, the choice $\bar{\lambda} = 0$ (essentially) recovers Theorem 1, since there is no approximation error. Taking $q \to 0$, we are thus left with the bound

$$\mathbb{E}\|\bar{f} - f^*\|_2^2] \;\lesssim\; \lambda\, \|f^*\|_{\mathcal{H}}^2 + \frac{\gamma(\lambda)\rho^2\tau_0^2}{N}, \tag{14}$$

where $\lesssim$ denotes an inequality up to constants. By inspection, this bound is roughly equivalent to Theorem 1; see in particular the decomposition (8). On the other hand, when the condition $f^* \in \mathcal{H}$ fails to hold, we can take $\bar{\lambda} = \lambda$, and then choose q to balance between the familiar approximation and estimation errors: we have

$$\mathbb{E}[\|\bar{f} - f^*\|_2^2] \lesssim \left(1 + \frac{1}{q}\right) \underbrace{\inf_{\|f\|_{\mathcal{H}} \leq R} \|f - f^*\|_2^2}_{\text{approximation}} \;+\; (1+q) \underbrace{\left(\frac{\gamma(\lambda)\rho^2\tau_\lambda^2}{N}\right)}_{\text{estimation}}. \tag{15}$$

Relative to Theorem 1, the condition (10) required to apply Theorem 2 involves constraints on the number m of subsampled data sets that are more restrictive. In particular,

when ignoring constants and logarithm terms, the quantity m may grow at rate $\sqrt{N/\gamma^2(\lambda)}$. By contrast, Theorem 1 allows m to grow as quickly as $N/\gamma^2(\lambda)$ (recall the remarks on $T_3(d)$ following Theorem 1 or look ahead to condition (28)). Thus—at least in our current analysis—generalizing to the case that $f^* \notin \mathcal{H}$ prevents us from dividing the data into finer subsets.

3.3 Some Consequences

We now turn to deriving some explicit consequences of our main theorems for specific classes of reproducing kernel Hilbert spaces. In each case, our derivation follows the broad outline given the the remarks following Theorem 1: we first choose the regularization parameter λ to balance the bias and variance terms, and then show, by comparison to known minimax lower bounds, that the resulting upper bound is optimal. Finally, we derive an upper bound on the number of subsampled data sets m for which the minimax optimal convergence rate can still be achieved. Throughout this section, we assume that $f^* \in \mathcal{H}$.

3.3.1 Finite-rank Kernels

Our first corollary applies to problems for which the kernel has finite rank r, meaning that its eigenvalues satisfy $\mu_j = 0$ for all $j > r$. Examples of such finite rank kernels include the linear kernel $K(x, x') = \langle x, x' \rangle_{\mathbb{R}^d}$, which has rank at most $r = d$; and the kernel $K(x, x) = (1 + x\, x')^m$ generating polynomials of degree m, which has rank at most $r = m+1$.

Corollary 3 *For a kernel with rank r, consider the output of the Fast-KRR algorithm with $\lambda = r/N$. Suppose that Assumption B and Assumptions A (or A′) hold, and that the number of processors m satisfy the bound*

$$m \leq c \frac{N^{\frac{k-4}{k-2}}}{r^{2\frac{k-1}{k-2}} \rho^{\frac{4k}{k-2}} \log^{\frac{k}{k-2}} r} \quad \textit{(Assumption A)} \qquad \textit{or} \qquad m \leq c \frac{N}{r^2 \rho^4 \log N} \quad \textit{(Assumption A′)},$$

where c is a universal (numerical) constant. For suitably large N, the mean-squared error is bounded as

$$\mathbb{E}\left[\left\|\bar{f} - f^*\right\|_2^2\right] = \mathcal{O}(1)\frac{\sigma^2 r}{N}. \tag{16}$$

For finite-rank kernels, the rate (16) is known to be minimax-optimal, meaning that there is a universal constant $c' > 0$ such that

$$\inf_{\widetilde{f}} \sup_{\|f^*\|_{\mathcal{H}} \leq 1} \mathbb{E}[\|\widetilde{f} - f^*\|_2^2] \geq c' \frac{r}{N}, \tag{17}$$

where the infimum ranges over all estimators $\widetilde{f}$ based on observing all N samples (and with no constraints on memory and/or computation). This lower bound follows from Theorem 2(a) of Raskutti et al. (2012) with $s = d = 1$.

3.3.2 Polynomially Decaying Eigenvalues

Our next corollary applies to kernel operators with eigenvalues that obey a bound of the form

$$\mu_j \leq C\, j^{-2\nu} \quad \text{for all } j = 1, 2, \ldots, \tag{18}$$

where C is a universal constant, and $\nu > 1/2$ parameterizes the decay rate. We note that equation (5) assumes a finite kernel trace $\operatorname{tr}(K) := \sum_{j=1}^{\infty} \mu_j$. Since $\operatorname{tr}(K)$ appears in Theorem 1, it is natural to use $\sum_{j=1}^{\infty} Cj^{-2\nu}$ as an upper bound on $\operatorname{tr}(K)$. This upper bound is finite if and only if $\nu > 1/2$.

Kernels with polynomial decaying eigenvalues include those that underlie for the Sobolev spaces with different orders of smoothness (e.g. Birman and Solomjak, 1967; Gu, 2002). As a concrete example, the first-order Sobolev kernel $K(x, x') = 1 + \min\{x, x'\}$ generates an RKHS of Lipschitz functions with smoothness $\nu = 1$. Other higher-order Sobolev kernels also exhibit polynomial eigendecay with larger values of the parameter ν.

Corollary 4 *For any kernel with ν-polynomial eigendecay* (18)*, consider the output of the Fast-KRR algorithm with $\lambda = (1/N)^{\frac{2\nu}{2\nu+1}}$. Suppose that Assumption B and Assumption A (or A′) hold, and that the number of processors satisfy the bound*

$$m \leq c\left(\frac{N^{\frac{2(k-4)\nu - k}{(2\nu+1)}}}{\rho^{4k}\log^k N}\right)^{\frac{1}{k-2}} \quad \textit{(Assumption A)} \quad \textit{or} \quad m \leq c\frac{N^{\frac{2\nu-1}{2\nu+1}}}{\rho^4 \log N} \quad \textit{(Assumption A′)},$$

where c is a constant only depending on ν. Then the mean-squared error is bounded as

$$\mathbb{E}\left[\left\|\bar{f} - f^*\right\|_2^2\right] = \mathcal{O}\left(\left(\frac{\sigma^2}{N}\right)^{\frac{2\nu}{2\nu+1}}\right). \tag{19}$$

The upper bound (19) is unimprovable up to constant factors, as shown by known minimax bounds on estimation error in Sobolev spaces (Stone, 1982; Tsybakov, 2009); see also Theorem 2(b) of Raskutti et al. (2012).

3.3.3 Exponentially Decaying Eigenvalues

Our final corollary applies to kernel operators with eigenvalues that obey a bound of the form

$$\mu_j \leq c_1 \exp(-c_2 j^2) \quad \text{for all } j = 1, 2, \ldots, \tag{20}$$

for strictly positive constants (c_1, c_2). Such classes include the RKHS generated by the Gaussian kernel $K(x, x') = \exp(-\|x - x'\|_2^2)$.

Corollary 5 *For a kernel with sub-Gaussian eigendecay* (20)*, consider the output of the Fast-KRR algorithm with $\lambda = 1/N$. Suppose that Assumption B and Assumption A (or A′) hold, and that the number of processors satisfy the bound*

$$m \leq c\frac{N^{\frac{k-4}{k-2}}}{\rho^{\frac{4k}{k-2}}\log^{\frac{2k-1}{k-2}} N} \quad \textit{(Assumption A)} \quad \textit{or} \quad m \leq c\frac{N}{\rho^4 \log^2 N} \quad \textit{(Assumption A′)},$$

where c is a constant only depending on c_2. Then the mean-squared error is bounded as

$$\mathbb{E}\left[\left\|\bar{f} - f^*\right\|_2^2\right] = \mathcal{O}\left(\sigma^2 \frac{\sqrt{\log N}}{N}\right). \tag{21}$$

The upper bound (21) is minimax optimal; see, for example, Theorem 1 and Example 2 of the recent paper by Yang et al. (2015).

3.3.4 Summary

Each corollary gives a critical threshold for the number m of data partitions: as long as m is below this threshold, the decomposition-based Fast-KRR algorithm gives the optimal rate of convergence. It is interesting to note that the number of splits may be quite large: each grows asymptotically with N whenever the basis functions have more than four moments (viz. Assumption A). Moreover, the Fast-KRR method can attain these optimal convergence rates while using substantially less computation than standard kernel ridge regression methods, as it requires solving problems only of size N/m.

3.4 The Choice of Regularization Parameter

In practice, the local sample size on each machine may be different and the optimal choice for the regularization λ may not be known *a priori*, so that an adaptive choice of the regularization parameter λ is desirable (e.g. Tsybakov, 2009, Chapters 3.5–3.7). We recommend using cross-validation to choose the regularization parameter, and we now sketch a heuristic argument that an adaptive algorithm using cross-validation may achieve optimal rates of convergence. (We leave fuller analysis to future work.)

Let λ_n be the (oracle) optimal regularization parameter given knowledge of the sampling distribution $\mathbb{P}$ and eigen-structure of the kernel K. We assume (cf. Corollary 4) that there is a constant $\nu > 0$ such that $\lambda_n \asymp n^{-\nu}$ as $n \to \infty$. Let n_i be the local sample size for each machine i and N the global sample size; we assume that $n_i \gg \sqrt{N}$ (clearly, $N \geq n_i$). First, use local cross-validation to choose regularization parameters $\widehat{\lambda}_{n_i}$ and $\widehat{\lambda}_{n_i^2/N}$ corresponding to samples of size n_i and n_i^2/N, respectively. Heuristically, if cross validation is successful, we expect to have $\widehat{\lambda}_{n_i} \simeq n_i^{-\nu}$ and $\widehat{\lambda}_{n_i^2/N} \simeq N^{\nu} n_i^{-2\nu}$, yielding that $\widehat{\lambda}_{n_i}^2 / \widehat{\lambda}_{n_i^2/N} \simeq N^{-\nu}$. With this intuition, we then compute local estimates

$$\widehat{f}_i := \operatorname*{argmin}_{f \in \mathcal{H}} \frac{1}{n_i} \sum_{(x,y) \in S_i} (f(x) - y)^2 + \widehat{\lambda}_{(i)} \|f\|_{\mathcal{H}}^2 \quad \text{where } \widehat{\lambda}_{(i)} := \frac{\widehat{\lambda}_{n_i}^2}{\widehat{\lambda}_{n_i^2/N}} \tag{22}$$

and global average estimate $\bar{f} = \sum_{i=1}^m \frac{n_i}{N} \widehat{f}_i$ as usual. Notably, we have $\widehat{\lambda}_{(i)} \simeq \lambda_N$ in this heuristic setting. Using formula (22) and the average $\bar{f}$, we have

$$\begin{aligned}
\mathbb{E}\Big[\left\|\bar{f} - f^*\right\|_2^2\Big] &= \mathbb{E}\left[\left\|\sum_{i=1}^m \frac{n_i}{N}\Big(\widehat{f}_i - \mathbb{E}[\widehat{f}_i]\Big)\right\|_2^2\right] + \left\|\sum_{i=1}^m \frac{n_i}{N}\Big(\mathbb{E}[\widehat{f}_i] - f^*\Big)\right\|_2^2 \\
&\leq \sum_{i=1}^m \frac{n_i^2}{N^2} \mathbb{E}\left[\left\|\widehat{f}_i - \mathbb{E}[\widehat{f}_i]\right\|_2^2\right] + \max_{i \in [m]} \left\{\left\|\mathbb{E}[\widehat{f}_i] - f^*\right\|_2^2\right\}. \qquad (23)
\end{aligned}$$

Using Lemmas 6 and 7 from the proof of Theorem 1 to come and assuming that $\widehat{\lambda}_n$ is concentrated tightly enough around λ_n, we obtain $\|\mathbb{E}[\widehat{f}_i] - f^*\|_2^2 = \mathcal{O}(\lambda_N \|f^*\|_{\mathcal{H}}^2)$ by Lemma 6 and that $\mathbb{E}[\|\widehat{f}_i - \mathbb{E}[\widehat{f}_i]\|_2^2] = \mathcal{O}(\frac{\gamma(\lambda_N)}{n_i})$ by Lemma 7. Substituting these bounds into inequality (23) and noting that $\sum_i n_i = N$, we may upper bound the overall estimation error as

$$\mathbb{E}\Big[\big\|\bar{f} - f^*\big\|_2^2 \Big] \leq \mathcal{O}(1) \cdot \left(\lambda_N \|f^*\|_{\mathcal{H}}^2 + \frac{\gamma(\lambda_N)}{N} \right).$$

While the derivation of this upper bound was non-rigorous, we believe that it is roughly accurate, and in comparison with the previous upper bound (8), it provides optimal rates of convergence.

4. Proofs of Theorem 1 and Related Results

We now turn to the proofs of Theorem 1 and Corollaries 3 through 5. This section contains only a high-level view of proof of Theorem 1; we defer more technical aspects to the appendices.

4.1 Proof of Theorem 1

Using the definition of the averaged estimate $\bar{f} = \frac{1}{m}\sum_{i=1}^m \widehat{f}_i$, a bit of algebra yields

$$\begin{aligned}
\mathbb{E}[\big\|\bar{f} - f^*\big\|_2^2] &= \mathbb{E}[\big\|(\bar{f} - \mathbb{E}[\bar{f}]) + (\mathbb{E}[\bar{f}] - f^*)\big\|_2^2] \\
&= \mathbb{E}[\big\|\bar{f} - \mathbb{E}[\bar{f}]\big\|_2^2] + \big\|\mathbb{E}[\bar{f}] - f^*\big\|_2^2 + 2\mathbb{E}[\langle \bar{f} - \mathbb{E}[\bar{f}], \mathbb{E}[\bar{f}] - f^* \rangle_{L^2(\mathbb{P})}] \\
&= \mathbb{E}\left[\left\| \frac{1}{m} \sum_{i=1}^m (\widehat{f}_i - \mathbb{E}[\widehat{f}_i]) \right\|_2^2 \right] + \big\|\mathbb{E}[\bar{f}] - f^*\big\|_2^2,
\end{aligned}$$

where we used the fact that $\mathbb{E}[\widehat{f}_i] = \mathbb{E}[\bar{f}]$ for each $i \in [m]$. Using this unbiasedness once more, we bound the variance of the terms $\widehat{f}_i - \mathbb{E}[\bar{f}]$ to see that

$$\begin{aligned}
\mathbb{E}\left[\big\|\bar{f} - f^*\big\|_2^2 \right] &= \frac{1}{m} \mathbb{E}\left[\|\widehat{f}_1 - \mathbb{E}[\widehat{f}_1]\|_2^2 \right] + \|\mathbb{E}[\widehat{f}_1] - f^*\|_2^2 \\
&\leq \frac{1}{m} \mathbb{E}\left[\|\widehat{f}_1 - f^*\|_2^2 \right] + \|\mathbb{E}[\widehat{f}_1] - f^*\|_2^2, \qquad (24)
\end{aligned}$$

where we have used the fact that $\mathbb{E}[\widehat{f}_i]$ minimizes $\mathbb{E}[\|\widehat{f}_i - f\|_2^2]$ over $f \in \mathcal{H}$.

The error bound (24) suggests our strategy: we upper bound $\mathbb{E}[\|\widehat{f}_1 - f^*\|_2^2]$ and $\|\mathbb{E}[\widehat{f}_1] - f^*\|_2^2$ respectively. Based on equation (3), the estimate $\widehat{f}_1$ is obtained from a standard kernel ridge regression with sample size $n = N/m$ and ridge parameter λ. Accordingly, the following two auxiliary results provide bounds on these two terms, where the reader should recall the definitions of $b(n, d, k)$ and β_d from equation (5). In each lemma, C represents a universal (numerical) constant.

Lemma 6 (Bias bound) *Under Assumptions A and B, for each $d = 1, 2, \ldots$, we have*

$$\|\mathbb{E}[\widehat{f}] - f^*\|_2^2 \leq 8\lambda \|f^*\|_{\mathcal{H}}^2 + \frac{8\rho^4 \|f^*\|_{\mathcal{H}}^2 \operatorname{tr}(K)\beta_d}{\lambda} + \left(Cb(n, d, k) \frac{\rho^2 \gamma(\lambda)}{\sqrt{n}} \right)^k \mu_0 \|f^*\|_{\mathcal{H}}^2. \qquad (25)$$

Lemma 7 (Variance bound) *Under Assumptions A and B, for each $d = 1, 2, \ldots$, we have*

$$\mathbb{E}[\|\widehat{f} - f^*\|_2^2] \leq 12\lambda \|f^*\|_{\mathcal{H}}^2 + \frac{12\sigma^2\gamma(\lambda)}{n} + \left(\frac{2\sigma^2}{\lambda} + 4\|f^*\|_{\mathcal{H}}^2\right)\left(\mu_{d+1} + \frac{12\rho^4 \operatorname{tr}(K)\beta_d}{\lambda} + \left(Cb(n,d,k)\frac{\rho^2\gamma(\lambda)}{\sqrt{n}}\right)^k \|f^*\|_2^2\right). \quad (26)$$

The proofs of these lemmas, contained in Appendices A and B respectively, constitute one main technical contribution of this paper. Given these two lemmas, the remainder of the theorem proof is straightforward. Combining the inequality (24) with Lemmas 6 and 7 yields the claim of Theorem 1.

Remarks: The proofs of Lemmas 6 and 7 are somewhat complex, but to the best of our knowledge, existing literature does not yield significantly simpler proofs. We now discuss this claim to better situate our technical contributions. Define the regularized population minimizer $f_\lambda^* := \operatorname{argmin}_{f\in\mathcal{H}}\{\mathbb{E}[(f(X)-Y)^2] + \lambda\|f\|_{\mathcal{H}}^2\}$. Expanding the decomposition (24) of the $L^2(\mathbb{P})$-risk into bias and variance terms, we obtain the further bound

$$\mathbb{E}\left[\left\|\bar{f} - f^*\right\|_2^2\right] \leq \|\mathbb{E}[\widehat{f}_1] - f^*\|_2^2 + \frac{1}{m}\mathbb{E}\left[\|\widehat{f}_1 - f^*\|_2^2\right]$$
$$= \underbrace{\|\mathbb{E}[\widehat{f}_1] - f^*\|_2^2}_{:=T_1} + \frac{1}{m}\Big(\underbrace{\|f_\lambda^* - f^*\|_2^2}_{:=T_2} + \underbrace{\mathbb{E}\left[\|\widehat{f}_1 - f^*\|_2^2\right] - \|f_\lambda^* - f^*\|_2^2}_{:=T_3}\Big) = T_1 + \frac{1}{m}(T_2 + T_3).$$

In this decomposition, T_1 and T_2 are bias and approximation error terms induced by the regularization parameter λ, while T_3 is an excess risk (variance) term incurred by minimizing the empirical loss.

This upper bound illustrates three trade-offs in our subsampled and averaged kernel regression procedure:

- The trade-off between T_2 and T_3: when the regularization parameter λ grows, the bias term T_2 increases while the variance term T_3 converges to zero.
- The trade-off between T_1 and T_3: when the regularization parameter λ grows, the bias term T_1 increases while the variance term T_3 converges to zero.
- The trade-off between T_1 and the computation time: when the number of machines m grows, the bias term T_1 increases (as the local sample size $n = N/m$ shrinks), while the computation time N^3/m^2 decreases.

Theoretical results in the KRR literature focus on the trade-off between T_2 and T_3, but in the current context, we also need an upper bound on the bias term T_1, which is not relevant for classical (centralized) analyses.

With this setting in mind, Lemma 6 tightly upper bounds the bias T_1 as a function of λ and n. An essential part of the proof is to characterize the properties of $\mathbb{E}[\widehat{f}_1]$, which is the expectation of a nonparametric empirical loss minimizer. We are not aware of existing

literature on this problem, and the proof of Lemma 6 introduces novel techniques for this purpose.

On the other hand, Lemma 7 upper bounds $\mathbb{E}[\|\widehat{f}_1 - f^*\|_2^2]$ as a function of λ and n. Past work has focused on bounding a quantity of this form, but for technical reasons, most work (e.g. van de Geer, 2000; Mendelson, 2002b; Bartlett et al., 2002; Zhang, 2005) focuses on analyzing the constrained form

$$\widehat{f}_i := \operatorname*{argmin}_{\|f\|_{\mathcal{H}} \leq C} \frac{1}{|S_i|} \sum_{(x,y) \in S_i} (f(x) - y)^2, \tag{27}$$

of kernel ridge regression. While this problem traces out the same set of solutions as that of the regularized kernel ridge regression estimator (3), it is non-trivial to determine a matched setting of λ for a given C. Zhang (2003) provides one of the few analyses of the regularized ridge regression estimator (3) (or (2)), providing an upper bound of the form $\mathbb{E}[\|\widehat{f} - f^*\|_2^2] = \mathcal{O}(\lambda + \frac{1/\lambda}{n})$, which is at best $\mathcal{O}(\frac{1}{\sqrt{n}})$. In contrast, Lemma 7 gives upper bound $\mathcal{O}(\lambda + \frac{\gamma(\lambda)}{n})$; the effective dimension $\gamma(\lambda)$ is often much smaller than $1/\lambda$, yielding a stronger convergence guarantee.

4.2 Proof of Corollary 3

We first present a general inequality bounding the size of m for which optimal convergence rates are possible. We assume that d is chosen large enough such that we have $\log(d) \geq k$ and $d \geq N$. In the rest of the proof, our assignment to d will satisfy these inequalities. In this case, inspection of Theorem 1 shows that if m is small enough that

$$\left(\sqrt{\frac{\log d}{N/m}}\rho^2\gamma(\lambda)\right)^k \frac{1}{m\lambda} \leq \frac{\gamma(\lambda)}{N},$$

then the term $T_3(d)$ provides a convergence rate given by $\gamma(\lambda)/N$. Thus, solving the expression above for m, we find

$$\frac{m \log d}{N}\rho^4\gamma(\lambda)^2 = \frac{\lambda^{2/k} m^{2/k} \gamma(\lambda)^{2/k}}{N^{2/k}} \quad \text{or} \quad m^{\frac{k-2}{k}} = \frac{\lambda^{\frac{2}{k}} N^{\frac{k-2}{k}}}{\gamma(\lambda)^{2\frac{k-1}{k}} \rho^4 \log d}.$$

Taking $(k-2)/k$-th roots of both sides, we obtain that if

$$m \leq \frac{\lambda^{\frac{2}{k-2}} N}{\gamma(\lambda)^{2\frac{k-1}{k-2}} \rho^{\frac{4k}{k-2}} \log^{\frac{k}{k-2}} d}, \tag{28}$$

then the term $T_3(d)$ of the bound (7) is $\mathcal{O}(\gamma(\lambda)/N)$.

Now we apply the bound (28) in the case in the corollary. Let us take $d = \max\{r, N\}$. Notice that $\beta_d = \beta_r = \mu_{r+1} = 0$. We find that $\gamma(\lambda) \leq r$ since each of its terms is bounded by 1, and we take $\lambda = r/N$. Evaluating the expression (28) with this value, we arrive at

$$m \leq \frac{N^{\frac{k-4}{k-2}}}{r^{2\frac{k-1}{k-2}} \rho^{\frac{4k}{k-2}} \log^{\frac{k}{k-2}} d}.$$

If we have sufficiently many moments that $k \geq \log N$, and $N \geq r$ (for example, if the basis functions ϕ_j have a uniform bound ρ, then k can be chosen arbitrarily large), then we may take $k = \log N$, which implies that $N^{\frac{k-4}{k-2}} = \Omega(N)$, $r^{2\frac{k-1}{k-2}} = \mathcal{O}(r^2)$ and $\rho^{\frac{4k}{k-2}} = \mathcal{O}(\rho^4)$; and we replace $\log d$ with $\log N$. Then so long as

$$m \leq c \frac{N}{r^2 \rho^4 \log N}$$

for some constant $c > 0$, we obtain an identical result.

4.3 Proof of Corollary 4

We follow the program outlined in our remarks following Theorem 1. We must first choose λ on the order of $\gamma(\lambda)/N$. To that end, we note that setting $\lambda = N^{-\frac{2\nu}{2\nu+1}}$ gives

$$\begin{aligned}\gamma(\lambda) = \sum_{j=1}^{\infty} \frac{1}{1 + j^{2\nu} N^{-\frac{2\nu}{2\nu+1}}} &\leq N^{\frac{1}{2\nu+1}} + \sum_{j > N^{\frac{1}{2\nu+1}}} \frac{1}{1 + j^{2\nu} N^{-\frac{2\nu}{2\nu+1}}} \\ &\leq N^{\frac{1}{2\nu+1}} + N^{\frac{2\nu}{2\nu+1}} \int_{N^{\frac{1}{2\nu+1}}} \frac{1}{u^{2\nu}} du = N^{\frac{1}{2\nu+1}} + \frac{1}{2\nu - 1} N^{\frac{1}{2\nu+1}}.\end{aligned}$$

Dividing by N, we find that $\lambda \approx \gamma(\lambda)/N$, as desired. Now we choose the truncation parameter d. By choosing $d = N^t$ for some $t \in \mathbb{R}_+$, then we find that $\mu_{d+1} \lesssim N^{-2\nu t}$ and an integration yields $\beta_d \lesssim N^{-(2\nu-1)t}$. Setting $t = 3/(2\nu - 1)$ guarantees that $\mu_{d+1} \lesssim N^{-3}$ and $\beta_d \lesssim N^{-3}$; the corresponding terms in the bound (7) are thus negligible. Moreover, we have for any finite k that $\log d \gtrsim k$.

Applying the general bound (28) on m, we arrive at the inequality

$$m \leq c \frac{N^{-\frac{4\nu}{(2\nu+1)(k-2)}} N}{N^{\frac{2(k-1)}{(2\nu+1)(k-2)}} \rho^{\frac{4k}{k-2}} \log^{\frac{k}{k-2}} N} = c \frac{N^{\frac{2(k-4)\nu - k}{(2\nu+1)(k-2)}}}{\rho^{\frac{4k}{k-2}} \log^{\frac{k}{k-2}} N}.$$

Whenever this holds, we have convergence rate $\lambda = N^{-\frac{2\nu}{2\nu+1}}$. Now, let Assumption A′ hold. Then taking $k = \log N$, the above bound becomes (to a multiplicative constant factor) $N^{\frac{2\nu-1}{2\nu+1}}/\rho^4 \log N$ as claimed.

4.4 Proof of Corollary 5

First, we set $\lambda = 1/N$. Considering the sum $\gamma(\lambda) = \sum_{j=1}^{\infty} \mu_j/(\mu_j + \lambda)$, we see that for $j \leq \sqrt{(\log N)/c_2}$, the elements of the sum are bounded by 1. For $j > \sqrt{(\log N)/c_2}$, we make the approximation

$$\sum_{j \geq \sqrt{(\log N)/c_2}} \frac{\mu_j}{\mu_j + \lambda} \leq \frac{1}{\lambda} \sum_{j \geq \sqrt{(\log N)/c_2}} \mu_j \lesssim N \int_{\sqrt{(\log N)/c_2}}^{\infty} \exp(-c_2 t^2) dt = \mathcal{O}(1).$$

Thus we find that $\gamma(\lambda) + 1 \leq c\sqrt{\log N}$ for some constant c. By choosing $d = N^2$, we have that the tail sum and $(d+1)$-th eigenvalue both satisfy $\mu_{d+1} \leq \beta_d \lesssim c_2^{-1} N^{-4}$. As a consequence, all the terms involving β_d or μ_{d+1} in the bound (7) are negligible.

Recalling our inequality (28), we thus find that (under Assumption A), as long as the number of partitions m satisfies

$$m \leq c \frac{N^{\frac{k-4}{k-2}}}{\rho^{\frac{4k}{k-2}} \log^{\frac{2k-1}{k-2}} N},$$

the convergence rate of $\bar{f}$ to f^* is given by $\gamma(\lambda)/N \simeq \sqrt{\log N}/N$. Under the boundedness assumption A$'$, as we did in the proof of Corollary 3, we take $k = \log N$ in Theorem 1. By inspection, this yields the second statement of the corollary.

5. Proof of Theorem 2 and Related Results

In this section, we provide the proofs of Theorem 2, as well as the bound (13) based on the alternative form of the residual error. As in the previous section, we present a high-level proof, deferring more technical arguments to the appendices.

5.1 Proof of Theorem 2

We begin by stating and proving two auxiliary claims:

$$\mathbb{E}\left[(Y - f(X))^2\right] = \mathbb{E}\left[(Y - f^*(X))^2\right] + \|f - f^*\|_2^2 \quad \text{for any } f \in L^2(\mathbb{P}), \quad \text{and} \tag{29a}$$

$$f_{\bar{\lambda}}^* = \operatorname*{argmin}_{\|f\|_{\mathcal{H}} \leq R} \|f - f^*\|_2^2. \tag{29b}$$

Let us begin by proving equality (29a). By adding and subtracting terms, we have

$$\begin{aligned}\mathbb{E}\left[(Y - f^*(X))^2\right] &= \mathbb{E}\left[(Y - f^*(X))^2\right] + \|f - f^*\|_2^2 + 2\mathbb{E}[(f(X) - f^*(X))\mathbb{E}[Y - f^*(X) \mid X]] \\ &\stackrel{(i)}{=} \mathbb{E}\left[(Y - f^*(X))^2\right] + \|f - f^*\|_2^2,\end{aligned}$$

where equality (i) follows since the random variable $Y - f^*(X)$ is mean-zero given $X = x$.

For the second equality (29b), consider any function f in the RKHS that satisfies the bound $\|f\|_{\mathcal{H}} \leq R$. The definition of the minimizer $f_{\bar{\lambda}}^*$ guarantees that

$$\mathbb{E}\left[(f_{\bar{\lambda}}^*(X) - Y)^2\right] + \bar{\lambda} R^2 \leq \mathbb{E}[(f(X) - Y)^2] + \bar{\lambda}\|f\|_{\mathcal{H}}^2 \leq \mathbb{E}[(f(X) - Y)^2] + \bar{\lambda} R^2.$$

This result combined with equation (29a) establishes the equality (29b).

We now turn to the proof of the theorem. Applying Hölder's inequality yields that

$$\begin{aligned}\left\|\bar{f} - f^*\right\|_2^2 &\leq \left(1 + \frac{1}{q}\right)\left\|f_{\bar{\lambda}}^* - f^*\right\|_2^2 + (1+q)\left\|\bar{f} - f_{\bar{\lambda}}^*\right\|_2^2 \\ &= \left(1 + \frac{1}{q}\right)\inf_{\|f\|_{\mathcal{H}} \leq R}\|f - f^*\|_2^2 + (1+q)\left\|\bar{f} - f_{\bar{\lambda}}^*\right\|_2^2 \qquad \text{for all } q > 0, \end{aligned} \tag{30}$$

where the second step follows from equality (29b). It thus suffices to upper bound $\left\|\bar{f} - f_{\bar{\lambda}}^*\right\|_2^2$, and following the deduction of inequality (24), we immediately obtain the decomposition formula

$$\mathbb{E}\left[\left\|\bar{f} - f_{\bar{\lambda}}^*\right\|_2^2\right] \leq \frac{1}{m}\mathbb{E}[\|\widehat{f}_1 - f_{\bar{\lambda}}^*\|_2^2] + \|\mathbb{E}[\widehat{f}_1] - f_{\bar{\lambda}}^*\|_2^2, \tag{31}$$

where $\widehat{f}_1$ denotes the empirical minimizer for *one* of the subsampled datasets (i.e. the standard KRR solution on a sample of size $n = N/m$ with regularization λ). This suggests our strategy, which parallels our proof of Theorem 1: we upper bound $\mathbb{E}[\|\widehat{f}_1 - f^*_{\bar{\lambda}}\|_2^2]$ and $\|\mathbb{E}[\widehat{f}_1] - f^*_{\bar{\lambda}}\|_2^2$, respectively. In the rest of the proof, we let $\widehat{f} = \widehat{f}_1$ denote this solution.

Let the estimation error for a subsample be given by $\Delta = \widehat{f} - f^*_{\bar{\lambda}}$. Under Assumptions A and B′, we have the following two lemmas bounding expression (31), which parallel Lemmas 6 and 7 in the case when $f^* \in \mathcal{H}$. In each lemma, C denotes a universal constant.

Lemma 8 *For all $d = 1, 2, \ldots$, we have*

$$\mathbb{E}\left[\|\Delta\|_2^2\right] \leq \frac{16(\bar{\lambda}-\lambda)^2 R^2}{\lambda} + \frac{8\gamma(\lambda)\rho^2\tau^2_{\bar{\lambda}}}{n} + \sqrt{32R^4 + 8\tau^4_{\bar{\lambda}}/\lambda^2}\left(\mu_{d+1} + \frac{16\rho^4 \operatorname{tr}(K)\beta_d}{\lambda} + \left(Cb(n,d,k)\frac{\rho^2\gamma(\lambda)}{\sqrt{n}}\right)^k\right). \quad (32)$$

Denoting the right hand side of inequality (32) by D^2, we have

Lemma 9 *For all $d = 1, 2, \ldots$, we have*

$$\|\mathbb{E}[\Delta]\|_2^2 \leq \frac{4(\bar{\lambda}-\lambda)^2 R^2}{\lambda} + \frac{C\log^2(d)(\rho^2\gamma(\lambda))^2}{n}D^2 + \sqrt{32R^4 + 8\tau^4_{\bar{\lambda}}/\lambda^2}\left(\mu_{d+1} + \frac{4\rho^4 \operatorname{tr}(K)\beta_d}{\lambda}\right). \quad (33)$$

See Appendices C and D for the proofs of these two lemmas.

Given these two lemmas, we can now complete the proof of the theorem. If the conditions (10) hold, we have

$$\beta_d \leq \frac{\lambda}{(R^2 + \tau^2_{\bar{\lambda}}/\lambda)N}, \quad \mu_{d+1} \leq \frac{1}{(R^2 + \tau^2_{\bar{\lambda}}/\lambda)N},$$
$$\frac{\log^2(d)(\rho^2\gamma(\lambda))^2}{n} \leq \frac{1}{m} \quad \text{and} \quad \left(b(n,d,k)\frac{\rho^2\gamma(\lambda)}{\sqrt{n}}\right)^k \leq \frac{1}{(R^2 + \tau^2_{\bar{\lambda}}/\lambda)N},$$

so there is a universal constant C' satisfying

$$\sqrt{32R^4 + 8\tau^4_{\bar{\lambda}}/\lambda^2}\left(\mu_{d+1} + \frac{16\rho^4 \operatorname{tr}(K)\beta_d}{\lambda} + \left(Cb(n,d,k)\frac{\rho^2\gamma(\lambda)}{\sqrt{n}}\right)^k\right) \leq \frac{C'}{N}.$$

Consequently, Lemma 8 yields the upper bound

$$\mathbb{E}[\|\Delta\|_2^2] \leq \frac{8(\bar{\lambda}-\lambda)^2 R^2}{\lambda} + \frac{8\gamma(\lambda)\rho^2\tau^2_{\bar{\lambda}}}{n} + \frac{C'}{N}.$$

Since $\log^2(d)(\rho^2\gamma(\lambda))^2/n \leq 1/m$ by assumption, we obtain

$$\mathbb{E}\left[\|\bar{f} - f_\lambda^*\|_2^2\right] \leq \frac{C(\bar{\lambda}-\lambda)^2R^2}{\lambda m} + \frac{C\gamma(\lambda)\rho^2\tau_{\bar{\lambda}}^2}{N} + \frac{C}{Nm} + \frac{4(\bar{\lambda}-\lambda)^2R^2}{\lambda} + \frac{C(\bar{\lambda}-\lambda)^2R^2}{\lambda m} + \frac{C\gamma(\lambda)\rho^2\tau_{\bar{\lambda}}^2}{N} + \frac{C}{Nm} + \frac{C}{N},$$

where C is a universal constant (whose value is allowed to change from line to line). Summing these bounds and using the condition that $\lambda \geq \bar{\lambda}$, we conclude that

$$\mathbb{E}\left[\|\bar{f} - f_\lambda^*\|_2^2\right] \leq \left(4 + \frac{C}{m}\right)(\lambda - \bar{\lambda})R^2 + \frac{C\gamma(\lambda)\rho^2\tau_{\bar{\lambda}}^2}{N} + \frac{C}{N}.$$

Combining this error bound with inequality (30) completes the proof.

5.2 Proof of Bound (13)

Using Theorem 2, it suffices to show that

$$\tau_{\bar{\lambda}}^4 \leq 8\,\mathrm{tr}(K)^2\|f_{\bar{\lambda}}^*\|_{\mathcal{H}}^4\rho^4 + 8\kappa^4. \tag{34}$$

By the tower property of expectations and Jensen's inequality, we have

$$\tau_{\bar{\lambda}}^4 = \mathbb{E}[(\mathbb{E}[(f_{\bar{\lambda}}^*(x) - Y)^2 \mid X = x])^2] \leq \mathbb{E}[(f_{\bar{\lambda}}^*(X) - Y)^4] \;\leq\; 8\mathbb{E}[(f_{\bar{\lambda}}^*(X))^4] + 8\mathbb{E}[Y^4].$$

Since we have assumed that $\mathbb{E}[Y^4] \leq \kappa^4$, the only remaining step is to upper bound $\mathbb{E}[(f_{\bar{\lambda}}^*(X))^4]$. Let $f_{\bar{\lambda}}^*$ have expansion $(\theta_1, \theta_2, \ldots)$ in the basis $\{\phi_j\}$. For any $x \in \mathcal{X}$, Hölder's inequality applied with the conjugates $4/3$ and 4 implies the upper bound

$$f_{\bar{\lambda}}^*(x) = \sum_{j=1}^\infty (\mu_j^{1/4}\theta_j^{1/2})\frac{\theta_j^{1/2}\phi_j(x)}{\mu_j^{1/4}} \leq \left(\sum_{j=1}^\infty \mu_j^{1/3}\theta_j^{2/3}\right)^{3/4}\left(\sum_{j=1}^\infty \frac{\theta_j^2}{\mu_j}\phi_j^4(x)\right)^{1/4}. \tag{35}$$

Again applying Hölder's inequality—this time with conjugates $3/2$ and 3—to upper bound the first term in the product in inequality (35), we obtain

$$\sum_{j=1}^\infty \mu_j^{1/3}\theta_j^{2/3} = \sum_{j=1}^\infty \mu_j^{2/3}\left(\frac{\theta_j^2}{\mu_j}\right)^{1/3} \leq \left(\sum_{j=1}^\infty \mu_j\right)^{2/3}\left(\sum_{j=1}^\infty \frac{\theta_j^2}{\mu_j}\right)^{1/3} = \mathrm{tr}(K)^{2/3}\|f_{\bar{\lambda}}^*\|_{\mathcal{H}}^{2/3}. \tag{36}$$

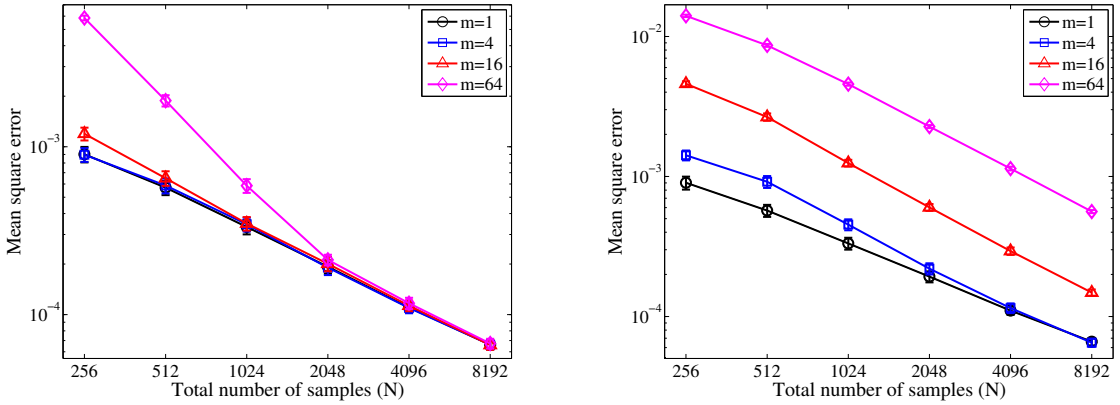
Combining inequalities (35) and (36), we find that

$$\mathbb{E}[(f_{\bar{\lambda}}^*(X))^4] \leq \mathrm{tr}(K)^2\|f_{\bar{\lambda}}^*\|_{\mathcal{H}}^2\sum_{j=1}^\infty \frac{\theta_j^2}{\mu_j}\mathbb{E}[\phi_j^4(X)] \leq \mathrm{tr}(K)^2\|f_{\bar{\lambda}}^*\|_{\mathcal{H}}^4\rho^4,$$

where we have used Assumption A. This completes the proof of inequality (34).

6. Experimental Results

In this section, we report the results of experiments on both simulated and real-world data designed to test the sharpness of our theoretical predictions.



(a) With under-regularization (b) Without under-regularization

Figure 1: The squared $L^2(\mathbb{P})$-norm between between the averaged estimate $\bar{f}$ and the optimal solution f^*. (a) These plots correspond to the output of the Fast-KRR algorithm: each sub-problem is under-regularized by using $\lambda \simeq N^{-2/3}$. (b) Analogous plots when each sub-problem is *not* under-regularized—that is, with $\lambda = n^{-2/3} = (N/m)^{-2/3}$ chosen as if there were only a single dataset of size n.

6.1 Simulation Studies

We begin by exploring the empirical performance of our subsample-and-average methods for a non-parametric regression problem on simulated datasets. For all experiments in this section, we simulate data from the regression model $y = f^*(x) + \varepsilon$ for $x \in [0,1]$, where $f^*(x) := \min(x, 1-x)$ is 1-Lipschitz, the noise variables $\varepsilon \sim \mathsf{N}(0, \sigma^2)$ are normally distributed with variance $\sigma^2 = 1/5$, and the samples $x_i \sim \mathsf{Uni}[0,1]$. The Sobolev space of Lipschitz functions on $[0,1]$ has reproducing kernel $K(x,x') = 1 + \min\{x, x'\}$ and norm $\|f\|_{\mathcal{H}}^2 = f^2(0) + \int_0^1 (f'(z))^2 dz$. By construction, the function $f^*(x) = \min(x, 1-x)$ satisfies $\|f^*\|_{\mathcal{H}} = 1$. The kernel ridge regression estimator $\widehat{f}$ takes the form

$$\widehat{f} = \sum_{i=1}^{N} \alpha_i K(x_i, \cdot), \quad \text{where} \quad \alpha = (K + \lambda N I)^{-1} y, \tag{37}$$

and K is the $N \times N$ Gram matrix and I is the $N \times N$ identity matrix. Since the first-order Sobolev kernel has eigenvalues (Gu, 2002) that scale as $\mu_j \simeq (1/j)^2$, the minimax convergence rate in terms of squared $L^2(\mathbb{P})$-error is $N^{-2/3}$ (see e.g. Tsybakov (2009); Stone (1982); Caponnetto and De Vito (2007)).

By Corollary 4 with $\nu = 1$, this optimal rate of convergence can be achieved by Fast-KRR with regularization parameter $\lambda \approx N^{-2/3}$ as long as the number of partitions m satisfies $m \lesssim N^{1/3}$. In each of our experiments, we begin with a dataset of size $N = mn$, which we partition uniformly at random into m disjoint subsets. We compute the local estimator $\widehat{f}_i$ for each of the m subsets using n samples via (37), where the Gram matrix is constructed using the ith batch of samples (and n replaces N). We then compute $\bar{f} = (1/m) \sum_{i=1}^{m} \widehat{f}_i$.

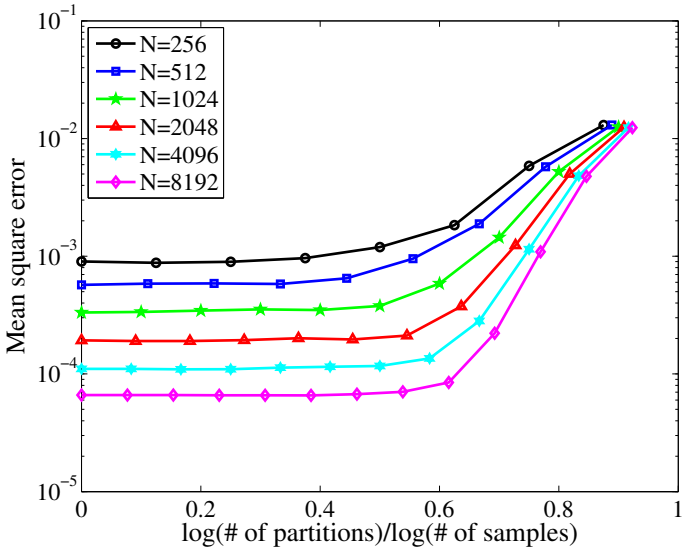


Figure 2: The mean-square error curves for fixed sample size but varied number of partitions. We are interested in the threshold of partitioning number m under which the optimal rate of convergence is achieved.

Our experiments compare the error of $\bar{f}$ as a function of sample size N, the number of partitions m, and the regularization λ.

In Figure 6.1(a), we plot the error $\|\bar{f} - f^*\|_2^2$ versus the total number of samples N, where $N \in \{2^8, 2^9, \ldots, 2^{13}\}$, using four different data partitions $m \in \{1, 4, 16, 64\}$. We execute each simulation 20 times to obtain standard errors for the plot. The black circled curve ($m = 1$) gives the baseline KRR error; if the number of partitions $m \leq 16$, Fast-KRR has accuracy comparable to the baseline algorithm. Even with $m = 64$, Fast-KRR's performance closely matches the full estimator for larger sample sizes ($N \geq 2^{11}$). In the right plot Figure 6.1(b), we perform an identical experiment, but we over-regularize by choosing $\lambda = n^{-2/3}$ rather than $\lambda = N^{-2/3}$ in each of the m sub-problems, combining the local estimates by averaging as usual. In contrast to Figure 6.1(a), there is an obvious gap between the performance of the algorithms when $m = 1$ and $m > 1$, as our theory predicts.

It is also interesting to understand the number of partitions m into which a dataset of size N may be divided while maintaining good statistical performance. According to Corollary 4 with $\nu = 1$, for the first-order Sobolev kernel, performance degradation should be limited as long as $m \lesssim N^{1/3}$. In order to test this prediction, Figure 2 plots the mean-square error $\|\bar{f} - f^*\|_2^2$ versus the ratio $\log(m)/\log(N)$. Our theory predicts that even as the number of partitions m may grow polynomially in N, the error should grow only above some constant value of $\log(m)/\log(N)$. As Figure 2 shows, the point that $\|\bar{f} - f^*\|_2$ begins to increase appears to be around $\log(m) \approx 0.45 \log(N)$ for reasonably large N. This empirical performance is somewhat better than the $(1/3)$ thresholded predicted by Corollary 4, but it does confirm that the number of partitions m can scale polynomially with N while retaining minimax optimality.

N		$m = 1$	$m = 16$	$m = 64$	$m = 256$	$m = 1024$
2^{12}	Error	$1.26 \cdot 10^{-4}$	$1.33 \cdot 10^{-4}$	$1.38 \cdot 10^{-4}$	N/A	N/A
	Time	1.12 (0.03)	0.03 (0.01)	0.02 (0.00)		
2^{13}	Error	$6.40 \cdot 10^{-5}$	$6.29 \cdot 10^{-5}$	$6.72 \cdot 10^{-5}$	N/A	N/A
	Time	5.47 (0.22)	0.12 (0.03)	0.04 (0.00)		
2^{14}	Error	$3.95 \cdot 10^{-5}$	$4.06 \cdot 10^{-5}$	$4.03 \cdot 10^{-5}$	$3.89 \cdot 10^{-5}$	N/A
	Time	30.16 (0.87)	0.59 (0.11)	0.11 (0.00)	0.06 (0.00)	
2^{15}	Error	Fail	$2.90 \cdot 10^{-5}$	$2.84 \cdot 10^{-5}$	$2.78 \cdot 10^{-5}$	N/A
	Time		2.65 (0.04)	0.43 (0.02)	0.15 (0.01)	
2^{16}	Error	Fail	$1.75 \cdot 10^{-5}$	$1.73 \cdot 10^{-5}$	$1.71 \cdot 10^{-5}$	$1.67 \cdot 10^{-5}$
	Time		16.65 (0.30)	2.21 (0.06)	0.41 (0.01)	0.23 (0.01)
2^{17}	Error	Fail	$1.19 \cdot 10^{-5}$	$1.21 \cdot 10^{-5}$	$1.25 \cdot 10^{-5}$	$1.24 \cdot 10^{-5}$
	Time		90.80 (3.71)	10.87 (0.19)	1.88 (0.08)	0.60 (0.02)

Table 1: Timing experiment giving $\|\bar{f} - f^*\|_2^2$ as a function of number of partitions m and data size N, providing mean run-time (measured in second) for each number m of partitions and data size N.

Our final experiment gives evidence for the improved time complexity partitioning provides. Here we compare the amount of time required to solve the KRR problem using the naive matrix inversion (37) for different partition sizes m and provide the resulting squared errors $\|\bar{f} - f^*\|_2^2$. Although there are more sophisticated solution strategies, we believe this is a reasonable proxy to exhibit Fast-KRR's potential. In Table 1, we present the results of this simulation, which we performed in Matlab using a Windows machine with 16GB of memory and a single-threaded 3.4Ghz processor. In each entry of the table, we give the mean error of Fast-KRR and the mean amount of time it took to run (with standard deviation over 10 simulations in parentheses; the error rate standard deviations are an order of magnitude smaller than the errors, so we do not report them). The entries "Fail" correspond to out-of-memory failures because of the large matrix inversion, while entries "N/A" indicate that $\|\bar{f} - f^*\|_2$ was significantly larger than the optimal value (rendering time improvements meaningless). The table shows that without sacrificing accuracy, decomposition via Fast-KRR can yield substantial computational improvements.

6.2 Real Data Experiments

We now turn to the results of experiments studying the performance of Fast-KRR on the task of predicting the year in which a song was released based on audio features associated with the song. We use the Million Song Dataset (Bertin-Mahieux et al., 2011), which consists of 463,715 training examples and a second set of 51,630 testing examples. Each example is a song (track) released between 1922 and 2011, and the song is represented as a vector of timbre information computed about the song. Each sample consists of the pair $(x_i, y_i) \in \mathbb{R}^d \times \mathbb{R}$, where $x_i \in \mathbb{R}^d$ is a $d = 90$-dimensional vector and $y_i \in [1922, 2011]$ is the year in which the song was released. (For further details, see Bertin-Mahieux et al. (2011)).

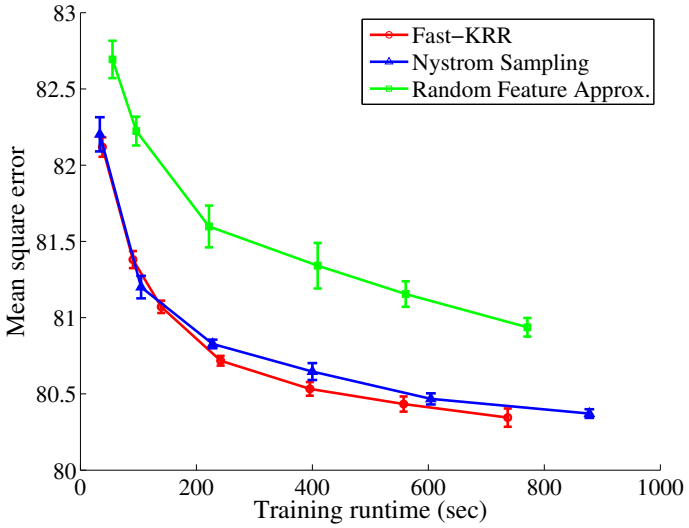


Figure 3: Results on year prediction on held-out test songs for Fast-KRR, Nyström sampling, and random feature approximation. Error bars indicate standard deviations over ten experiments.

Our experiments with this dataset use the Gaussian radial basis kernel

$$K(x, x') = \exp\left(-\frac{\|x - x'\|_2^2}{2\sigma^2}\right). \tag{38}$$

We normalize the feature vectors x so that the timbre signals have standard deviation 1, and select the bandwidth parameter $\sigma = 6$ via cross-validation. For regularization, we set $\lambda = N^{-1}$; since the Gaussian kernel has exponentially decaying eigenvalues (for typical distributions on X), Corollary 5 shows that this regularization achieves the optimal rate of convergence for the Hilbert space.

In Figure 3, we compare the time-accuracy curve of Fast-KRR with two approximation-based methods, plotting the mean-squared error between the predicted release year and the actual year on test songs. The first baseline is Nyström subsampling (Williams and Seeger, 2001), where the kernel matrix is approximated by a low-rank matrix of rank $r \in \{1, \ldots, 6\} \times 10^3$. The second baseline approach is an approximate form of kernel ridge regression using random features (Rahimi and Recht, 2007). The algorithm approximates the Gaussian kernel (38) by the inner product of two random feature vectors of dimensions $D \in \{2, 3, 5, 7, 8.5, 10\} \times 10^3$, and then solves the resulting linear regression problem. For the Fast-KRR algorithm, we use seven partitions $m \in \{32, 38, 48, 64, 96, 128, 256\}$ to test the algorithm. Each algorithm is executed 10 times to obtain standard deviations (plotted as error-bars in Figure 3).

As we see in Figure 3, for a fixed time budget, Fast-KRR enjoys the best performance, though the margin between Fast-KRR and Nyström sampling is not substantial. In spite of this close performance between Nyström sampling and the divide-and-conquer Fast-KRR

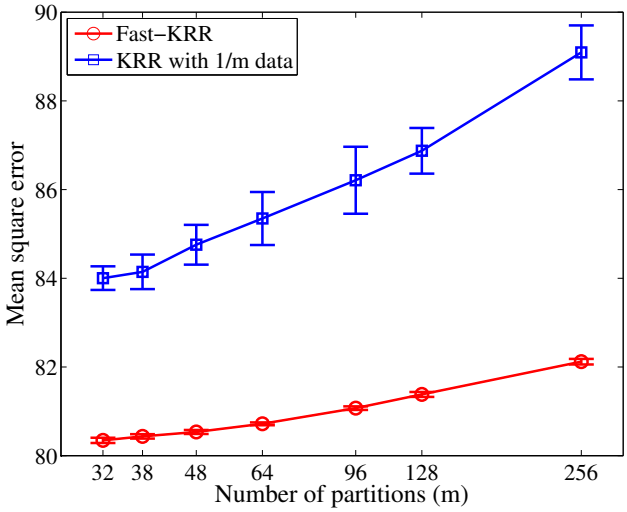


Figure 4: Comparison of the performance of Fast-KRR to a standard KRR estimator using a fraction $1/m$ of the data.

algorithm, it is worth noting that with parallel computation, it is trivial to accelerate Fast-KRR m times; parallelizing approximation-based methods appears to be a non-trivial task. Moreover, as our results in Section 3 indicate, Fast-KRR is minimax optimal in many regimes. At the same time the conference version of this paper was submitted, Bach (2013) published the first results we know of establishing convergence results in ℓ_2-error for Nyström sampling; see the discussion for more detail. We note in passing that standard linear regression with the original 90 features, while quite fast with runtime on the order of 1 second (ignoring data loading), has mean-squared-error 90.44, which is significantly worse than the kernel-based methods.

Our final experiment provides a sanity check: is the final averaging step in Fast-KRR even necessary? To this end, we compare Fast-KRR with standard KRR using a fraction $1/m$ of the data. For the latter approach, we employ the standard regularization $\lambda \approx (N/m)^{-1}$. As Figure 4 shows, Fast-KRR achieves much lower error rates than KRR using only a fraction of the data. Moreover, averaging stabilizes the estimators: the standard deviations of the performance of Fast-KRR are negligible compared to those for standard KRR.

7. Discussion

In this paper, we present results establishing that our decomposition-based algorithm for kernel ridge regression achieves minimax optimal convergence rates whenever the number of splits m of the data is not too large. The error guarantees of our method depend on the *effective dimensionality* $\gamma(\lambda) = \sum_{j=1}^{\infty} \mu_j/(\mu_j + \lambda)$ of the kernel. For any number of splits

$m \lesssim N/\gamma(\lambda)^2$, our method achieves estimation error decreasing as

$$\mathbb{E}\left[\|\bar{f} - f^*\|_2^2\right] \lesssim \lambda \|f^*\|_{\mathcal{H}}^2 + \frac{\sigma^2 \gamma(\lambda)}{N}.$$

(In particular, recall the bound (8) following Theorem 1). Notably, this convergence rate is minimax optimal, and we achieve substantial computational benefits from the subsampling schemes, in that computational cost scales (nearly) linearly in N.

It is also interesting to consider the number of kernel evaluations required to implement our method. Our estimator requires m sub-matrices of the full kernel (Gram) matrix, each of size $N/m \times N/m$. Since the method may use $m \asymp N/\gamma^2(\lambda)$ machines, in the best case, it requires at most $N\gamma^2(\lambda)$ kernel evaluations. By contrast, Bach (2013) shows that Nyström-based subsampling can be used to form an estimator within a constant factor of optimal as long as the number of N-dimensional subsampled columns of the kernel matrix scales roughly as the *marginal dimension* $\widetilde{\gamma}(\lambda) = N \left\|\mathrm{diag}(K(K + \lambda N I)^{-1})\right\|_\infty$. Consequently, using roughly $N\widetilde{\gamma}(\lambda)$ kernel evaluations, Nyström subsampling can achieve optimal convergence rates. These two scalings–namely, $N\gamma^2(\lambda)$ versus $N\widetilde{\gamma}(\lambda)$—are currently not comparable: in some situations, such as when the data is not compactly supported, $\widetilde{\gamma}(\lambda)$ can scale linearly with N, while in others it appears to scale roughly as the true effective dimensionality $\gamma(\lambda)$. A natural question arising from these lines of work is to understand the true optimal scaling for these different estimators: is one fundamentally better than the other? Are there natural computational tradeoffs that can be leveraged at large scale? As datasets grow substantially larger and more complex, these questions should become even more important, and we hope to continue to study them.

Acknowledgments

We thank Francis Bach for interesting and enlightening conversations on the connections between this work and his paper (Bach, 2013) and Yining Wang for pointing out a mistake in an earlier version of this manuscript. We also thank two reviewers for useful feedback and comments. JCD was partially supported by a National Defense Science and Engineering Graduate Fellowship (NDSEG) and a Facebook PhD fellowship. This work was partially supported by ONR MURI grant N00014-11-1-0688 to MJW.

Appendix A. Proof of Lemma 6

This appendix is devoted to the bias bound stated in Lemma 6. Let $X = \{x_i\}_{i=1}^n$ be shorthand for the design matrix, and define the error vector $\Delta = \widehat{f} - f^*$. By Jensen's inequality, we have $\|\mathbb{E}[\Delta]\|_2 \leq \mathbb{E}[\|\mathbb{E}[\Delta \mid X]\|_2]$, so it suffices to provide a bound on $\|\mathbb{E}[\Delta \mid X]\|_2$. Throughout this proof and the remainder of the paper, we represent the kernel evaluator by the function ξ_x, where $\xi_x := K(x, \cdot)$ and $f(x) = \langle \xi_x, f \rangle$ for any $f \in \mathcal{H}$. Using this notation, the estimate $\widehat{f}$ minimizes the empirical objective

$$\frac{1}{n} \sum_{i=1}^n \left(\langle \xi_{x_i}, f \rangle_{\mathcal{H}} - y_i\right)^2 + \lambda \|f\|_{\mathcal{H}}^2. \tag{39}$$

$\widehat{f}$	Empirical KRR minimizer based on n samples
f^*	Optimal function generating data, where $y_i = f^*(x_i) + \varepsilon_i$
Δ	Error $\widehat{f} - f^*$
ξ_x	RKHS evaluator $\xi_x := K(x, \cdot)$, so $\langle f, \xi_x \rangle = \langle \xi_x, f \rangle = f(x)$
$\widehat{\Sigma}$	Operator mapping $\mathcal{H} \to \mathcal{H}$ defined as the outer product $\widehat{\Sigma} := \frac{1}{n} \sum_{i=1}^n \xi_{x_i} \otimes \xi_{x_i}$, so that $\widehat{\Sigma} f = \frac{1}{n} \sum_{i=1}^n \langle \xi_{x_i}, f \rangle \xi_{x_i}$
ϕ_j	jth orthonormal basis vector for $L^2(\mathbb{P})$
δ_j	Basis coefficients of Δ or $\mathbb{E}[\Delta \mid X]$ (depending on context), i.e. $\Delta = \sum_{j=1}^\infty \delta_j \phi_j$
θ_j	Basis coefficients of f^*, i.e. $f^* = \sum_{j=1}^\infty \theta_j \phi_j$
d	Integer-valued truncation point
M	Diagonal matrix with $M = \operatorname{diag}(\mu_1, \ldots, \mu_d)$
Q	Diagonal matrix with $Q = (I_{d \times d} + \lambda M^{-1})^{\frac{1}{2}}$
Φ	$n \times d$ matrix with coordinates $\Phi_{ij} = \phi_j(x_i)$
$v^\downarrow$	Truncation of vector v. For $v = \sum_j \nu_j \phi_j \in \mathcal{H}$, defined as $v^\downarrow = \sum_{j=1}^d \nu_j \phi_j$; for $v \in \ell_2(\mathbb{N})$ defined as $v^\downarrow = (v_1, \ldots, v_d)$
$v^\uparrow$	Untruncated part of vector v, defined as $v^\uparrow = (v_{d+1}, v_{d+1}, \ldots)$
β_d	The tail sum $\sum_{j>d} \mu_j$
$\gamma(\lambda)$	The sum $\sum_{j=1}^\infty 1/(1 + \lambda/\mu_j)$
$b(n, d, k)$	The maximum $\max\{\sqrt{\max\{k, \log(d)\}}, \max\{k, \log(d)\}/n^{1/2-1/k}\}$

Table 2: Notation used in proofs

This objective is Fréchet differentiable, and as a consequence, the necessary and sufficient conditions for optimality (Luenberger, 1969) of $\widehat{f}$ are that

$$\frac{1}{n} \sum_{i=1}^n \xi_{x_i} (\langle \xi_{x_i}, \widehat{f} - f^* \rangle_{\mathcal{H}} - \varepsilon_i) + \lambda \widehat{f} = \frac{1}{n} \sum_{i=1}^n \xi_{x_i} (\langle \xi_{x_i}, \widehat{f} \rangle_{\mathcal{H}} - y_i) + \lambda \widehat{f} = 0, \tag{40}$$

where the last equation uses the fact that $y_i = \langle \xi_{x_i}, f^* \rangle_{\mathcal{H}} + \varepsilon_i$. Taking conditional expectations over the noise variables $\{\varepsilon_i\}_{i=1}^n$ with the design $X = \{x_i\}_{i=1}^n$ fixed, we find that

$$\frac{1}{n} \sum_{i=1}^n \xi_{x_i} \langle \xi_{x_i}, \mathbb{E}[\Delta \mid X] \rangle + \lambda \mathbb{E}[\widehat{f} \mid X] = 0.$$

Define the sample covariance operator $\widehat{\Sigma} := \frac{1}{n} \sum_{i=1}^n \xi_{x_i} \otimes \xi_{x_i}$. Adding and subtracting λf^* from the above equation yields

$$(\widehat{\Sigma} + \lambda I) \mathbb{E}[\Delta \mid X] = -\lambda f^*. \tag{41}$$

Consequently, we see we have $\|\mathbb{E}[\Delta \mid X]\|_{\mathcal{H}} \leq \|f^*\|_{\mathcal{H}}$, since $\widehat{\Sigma} \succeq 0$.

We now use a truncation argument to reduce the problem to a finite dimensional problem. To do so, we let $\delta \in \ell_2(\mathbb{N})$ denote the coefficients of $\mathbb{E}[\Delta \mid X]$ when expanded in the

basis $\{\phi_j\}_{j=1}^{\infty}$:

$$\mathbb{E}[\Delta \mid X] = \sum_{j=1}^{\infty} \delta_j \phi_j, \quad \text{with } \delta_j = \langle \mathbb{E}[\Delta \mid X], \phi_j \rangle_{L^2(\mathbb{P})}. \tag{42}$$

For a fixed $d \in \mathbb{N}$, define the vectors $\delta^{\downarrow} := (\delta_1, \ldots, \delta_d)$ and $\delta^{\uparrow} := (\delta_{d+1}, \delta_{d+2}, \ldots)$ (we suppress dependence on d for convenience). By the orthonormality of the collection $\{\phi_j\}$, we have

$$\|\mathbb{E}[\Delta \mid X]\|_2^2 = \|\delta\|_2^2 = \|\delta^{\downarrow}\|_2^2 + \|\delta^{\uparrow}\|_2^2. \tag{43}$$

We control each of the elements of the sum (43) in turn.

Control of the term $\|\delta^{\uparrow}\|_2^2$: By definition, we have

$$\|\delta^{\uparrow}\|_2^2 = \frac{\mu_{d+1}}{\mu_{d+1}} \sum_{j=d+1}^{\infty} \delta_j^2 \leq \mu_{d+1} \sum_{j=d+1}^{\infty} \frac{\delta_j^2}{\mu_j} \overset{(i)}{\leq} \mu_{d+1} \|\mathbb{E}[\Delta \mid X]\|_{\mathcal{H}}^2 \overset{(ii)}{\leq} \mu_{d+1} \|f^*\|_{\mathcal{H}}^2, \tag{44}$$

where inequality (i) follows since $\|\mathbb{E}[\Delta \mid X]\|_{\mathcal{H}}^2 = \sum_{j=1}^{\infty} \frac{\delta_j^2}{\mu_j}$; and inequality (ii) follows from the bound $\|\mathbb{E}[\Delta \mid X]\|_{\mathcal{H}} \leq \|f^*\|_{\mathcal{H}}$, which is a consequence of equality (41).

Control of the term $\|\delta^{\downarrow}\|_2^2$: Let $(\theta_1, \theta_2, \ldots)$ be the coefficients of f^* in the basis $\{\phi_j\}$. In addition, define the matrices $\Phi \in \mathbb{R}^{n \times d}$ by

$$\Phi_{ij} = \phi_j(x_i) \quad \text{for } i \in \{1, \ldots, n\}, \text{ and } j \in \{1, \ldots, d\}$$

and $M = \operatorname{diag}(\mu_1, \ldots, \mu_d) \succ 0 \in \mathbb{R}^{d \times d}$. Lastly, define the tail error vector $v \in \mathbb{R}^n$ by

$$v_i := \sum_{j > d} \delta_j \phi_j(x_i) = \mathbb{E}[\Delta^{\uparrow}(x_i) \mid X].$$

Let $l \in \mathbb{N}$ be arbitrary. Computing the (Hilbert) inner product of the terms in equation (41) with ϕ_l, we obtain

$$\begin{aligned} -\lambda \frac{\theta_l}{\mu_l} &= \langle \phi_l, -\lambda f^* \rangle = \left\langle \phi_l, (\widehat{\Sigma} + \lambda) \mathbb{E}[\Delta \mid X] \right\rangle \\ &= \frac{1}{n} \sum_{i=1}^{n} \langle \phi_l, \xi_{x_i} \rangle \langle \xi_{x_i}, \mathbb{E}[\Delta \mid X] \rangle + \lambda \langle \phi_l, \mathbb{E}[\Delta \mid X] \rangle = \frac{1}{n} \sum_{i=1}^{n} \phi_l(x_i) \mathbb{E}[\Delta(x_i) \mid X] + \lambda \frac{\delta_l}{\mu_l}. \end{aligned}$$

We can rewrite the final sum above using the fact that $\Delta = \Delta^{\downarrow} + \Delta^{\uparrow}$, which implies

$$\frac{1}{n} \sum_{i=1}^{n} \phi_l(x_i) \mathbb{E}[\Delta(x_i) \mid X] = \frac{1}{n} \sum_{i=1}^{n} \phi_l(x_i) \left(\sum_{j=1}^{d} \phi_j(x_i) \delta_j + \sum_{j > d} \phi_j(x_i) \delta_j \right)$$

Applying this equality for $l = 1, 2, \ldots, d$ yields

$$\left(\frac{1}{n} \Phi^T \Phi + \lambda M^{-1} \right) \delta^{\downarrow} = -\lambda M^{-1} \theta^{\downarrow} - \frac{1}{n} \Phi^T v. \tag{45}$$

We now show how the expression (45) gives us the desired bound in the lemma. By defining the shorthand matrix $Q = (I + \lambda M^{-1})^{1/2}$, we have

$$\frac{1}{n}\Phi^T\Phi + \lambda M^{-1} = I + \lambda M^{-1} + \frac{1}{n}\Phi^T\Phi - I = Q\left(I + Q^{-1}\left(\frac{1}{n}\Phi^T\Phi - I\right)Q^{-1}\right)Q.$$

As a consequence, we can rewrite expression (45) to

$$\left(I + Q^{-1}\left(\frac{1}{n}\Phi^T\Phi - I\right)Q^{-1}\right)Q\delta^{\downarrow} = -\lambda Q^{-1}M^{-1}\theta^{\downarrow} - \frac{1}{n}Q^{-1}\Phi^T v. \tag{46}$$

We now present a lemma bounding the terms in equality (46) to control $\delta^{\downarrow}$.

Lemma 10 *The following bounds hold:*

$$\left\|\lambda Q^{-1}M^{-1}\theta^{\downarrow}\right\|_2^2 \le \lambda \left\|f^*\right\|_{\mathcal{H}}^2, \quad \text{and} \tag{47a}$$

$$\mathbb{E}\left[\left\|\frac{1}{n}Q^{-1}\Phi^T v\right\|_2^2\right] \le \frac{\rho^4 \left\|f^*\right\|_{\mathcal{H}}^2 \operatorname{tr}(K)\beta_d}{\lambda}. \tag{47b}$$

Define the event $\mathcal{E} := \left\{\left\|\left|Q^{-1}\left(\frac{1}{n}\Phi^T\Phi - I\right)Q^{-1}\right|\right\| \le 1/2\right\}$. *Under Assumption A with moment bound* $\mathbb{E}[\phi_j(X)^{2k}] \le \rho^{2k}$, *there exists a universal constant* C *such that*

$$\mathbb{P}(\mathcal{E}^c) \le \left(\max\left\{\sqrt{k \vee \log(d)}, \frac{k \vee \log(d)}{n^{1/2-1/k}}\right\}\frac{C\rho^2\gamma(\lambda)}{\sqrt{n}}\right)^k. \tag{48}$$

We defer the proof of this lemma to Appendix A.1.

Based on this lemma, we can now complete the proof. Whenever the event $\mathcal{E}$ holds, we know that $I + Q^{-1}((1/n)\Phi^T\Phi - I)Q^{-1} \succeq (1/2)I$. In particular, we have

$$\left\|Q\delta^{\downarrow}\right\|_2^2 \le 4\left\|\lambda Q^{-1}M^{-1}\theta^{\downarrow} + (1/n)Q^{-1}\Phi^T v\right\|_2^2$$

on $\mathcal{E}$, by Eq. (46). Since $\|Q\delta^{\downarrow}\|_2^2 \ge \|\delta^{\downarrow}\|_2^2$, the above inequality implies that

$$\left\|\delta^{\downarrow}\right\|_2^2 \le 4\left\|\lambda Q^{-1}M^{-1}\theta^{\downarrow} + (1/n)Q^{-1}\Phi^T v\right\|_2^2$$

Since $\mathcal{E}$ is X-measurable, we thus obtain

$$\begin{aligned}\mathbb{E}\left[\left\|\delta^{\downarrow}\right\|_2^2\right] &= \mathbb{E}\left[1(\mathcal{E})\left\|\delta^{\downarrow}\right\|_2^2\right] + \mathbb{E}\left[1(\mathcal{E}^c)\left\|\delta^{\downarrow}\right\|_2^2\right] \\ &\le 4\mathbb{E}\left[1(\mathcal{E})\left\|\lambda Q^{-1}M^{-1}\theta^{\downarrow} + (1/n)Q^{-1}\Phi^T v\right\|_2^2\right] + \mathbb{E}\left[1(\mathcal{E}^c)\left\|\delta^{\downarrow}\right\|_2^2\right].\end{aligned}$$

Applying the bounds (47a) and (47b), along with the elementary inequality $(a+b)^2 \le 2a^2 + 2b^2$, we have

$$\mathbb{E}\left[\left\|\delta^{\downarrow}\right\|_2^2\right] \le 8\lambda\left\|f^*\right\|_{\mathcal{H}}^2 + \frac{8\rho^4\left\|f^*\right\|_{\mathcal{H}}^2\operatorname{tr}(K)\beta_d}{\lambda} + \mathbb{E}\left[1(\mathcal{E}^c)\left\|\delta^{\downarrow}\right\|_2^2\right]. \tag{49}$$

Now we use the fact that by the gradient optimality condition (41),

$$\|\mathbb{E}[\Delta \mid X]\|_2^2 \leq \mu_0 \|\mathbb{E}[\Delta \mid X]\|_{\mathcal{H}}^2 \leq \mu_0 \|f^*\|_{\mathcal{H}}^2$$

Recalling the shorthand (6) for $b(n, d, k)$, we apply the bound (48) to see

$$\mathbb{E}\left[1(\mathcal{E}^c) \|\delta^{\downarrow}\|_2^2\right] \leq \mathbb{P}(\mathcal{E}^c)\mu_0 \|f^*\|_{\mathcal{H}}^2 \leq \left(\frac{Cb(n,d,k)\rho^2\gamma(\lambda)}{\sqrt{n}}\right)^k \mu_0 \|f^*\|_{\mathcal{H}}^2$$

Combining this with the inequality (49), we obtain the desired statement of Lemma 6.

A.1 Proof of Lemma 10

Proof of bound (47a)*:* Beginning with the proof of the bound (47a), we have

$$\begin{aligned}\left\|Q^{-1}M^{-1}\theta^{\downarrow}\right\|_2^2 &= (\theta^{\downarrow})^T(M^2 + \lambda M)^{-1}\theta^{\downarrow} \\ &\leq (\theta^{\downarrow})^T(\lambda M)^{-1}\theta^{\downarrow} = \frac{1}{\lambda}(\theta^{\downarrow})^T M^{-1}\theta^{\downarrow} \leq \frac{1}{\lambda}\|f^*\|_{\mathcal{H}}^2 .\end{aligned}$$

Multiplying both sides by λ^2 gives the result.

Proof of bound (47b)*:* Next we turn to the proof of the bound (47b). We begin by re-writing $Q^{-1}\Phi^T v$ as the product of two components:

$$\frac{1}{n}Q^{-1}\Phi^T v = (M + \lambda I)^{-1/2}\left(\frac{1}{n}M^{1/2}\Phi^T v\right). \tag{50}$$

The first matrix is a diagonal matrix whose operator norm is bounded:

$$\left\|(M + \lambda I)^{-1/2}\right\| = \max_{j\in[d]} \frac{1}{\sqrt{\mu_j + \lambda}} \leq \frac{1}{\sqrt{\lambda}}. \tag{51}$$

For the second factor in the product (50), the analysis is a little more complicated. Let $\Phi_\ell = (\phi_l(x_1), \ldots, \phi_l(x_n))$ be the ℓth column of Φ. In this case,

$$\left\|M^{1/2}\Phi^T v\right\|_2^2 = \sum_{\ell=1}^d \mu_\ell(\Phi_\ell^T v)^2 \leq \sum_{\ell=1}^d \mu_\ell \|\Phi_\ell\|_2^2 \|v\|_2^2, \tag{52}$$

using the Cauchy-Schwarz inequality. Taking expectations with respect to the design $\{x_i\}_{i=1}^n$ and applying Hölder's inequality yields

$$\mathbb{E}[\|\Phi_\ell\|_2^2 \|v\|_2^2] \leq \sqrt{\mathbb{E}[\|\Phi_\ell\|_2^4]}\sqrt{\mathbb{E}[\|v\|_2^4]}.$$

We bound each of the terms in this product in turn. For the first, we have

$$\mathbb{E}[\|\Phi_\ell\|_2^4] = \mathbb{E}\left[\left(\sum_{i=1}^n \phi_\ell^2(X_i)\right)^2\right] = \mathbb{E}\left[\sum_{i,j=1}^n \phi_\ell^2(X_i)\phi_\ell^2(X_j)\right] \leq n^2\mathbb{E}[\phi_\ell^4(X_1)] \leq n^2\rho^4$$

since the X_i are i.i.d., $\mathbb{E}[\phi_\ell^2(X_1)] \leq \sqrt{\mathbb{E}[\phi_\ell^4(X_1)]}$, and $\mathbb{E}[\phi_\ell^4(X_1)] \leq \rho^4$ by assumption. Turning to the term involving v, we have

$$v_i^2 = \Big(\sum_{j>d} \delta_j \phi_j(x_i)\Big)^2 \leq \Big(\sum_{j>d} \frac{\delta_j^2}{\mu_j}\Big)\Big(\sum_{j>d} \mu_j \phi_j^2(x_i)\Big)$$

by Cauchy-Schwarz. As a consequence, we find

$$\begin{aligned}\mathbb{E}[\|v\|_2^4] = \mathbb{E}\Big[\Big(n\frac{1}{n}\sum_{i=1}^n v_i^2\Big)^2\Big] \leq n^2 \frac{1}{n}\sum_{i=1}^n \mathbb{E}[v_i^4] &\leq n\sum_{i=1}^n \mathbb{E}\Big[\Big(\sum_{j>d}\frac{\delta_j^2}{\mu_j}\Big)^2\Big(\sum_{j>d}\mu_j\phi_j^2(X_i)\Big)^2\Big] \\ &\leq n^2 \mathbb{E}\Big[\|\mathbb{E}[\Delta \mid X]\|_{\mathcal{H}}^4 \Big(\sum_{j>d}\mu_j\phi_j^2(X_1)\Big)^2\Big],\end{aligned}$$

since the X_i are i.i.d. Using the fact that $\|\mathbb{E}[\Delta \mid X]\|_{\mathcal{H}} \leq \|f^*\|_{\mathcal{H}}$, we expand the second square to find

$$\frac{1}{n^2}\mathbb{E}[\|v\|_2^4] \leq \|f^*\|_{\mathcal{H}}^4 \sum_{j,k>d} \mathbb{E}\left[\mu_j\mu_k\phi_j^2(X_1)\phi_k^2(X_1)\right] \leq \|f^*\|_{\mathcal{H}}^4 \rho^4 \sum_{j,k>d}\mu_j\mu_k = \|f^*\|_{\mathcal{H}}^4\rho^4\Big(\sum_{j>d}\mu_j\Big)^2.$$

Combining our bounds on $\|\Phi_\ell\|_2$ and $\|v\|_2$ with our initial bound (52), we obtain the inequality

$$\mathbb{E}\left[\left\|M^{1/2}\Phi^T v\right\|_2^2\right] \leq \sum_{l=1}^d \mu_\ell \sqrt{n^2\rho^4}\sqrt{n^2\|f^*\|_{\mathcal{H}}^4\rho^4\Big(\sum_{j>d}\mu_j\Big)^2} = n^2\rho^4\|f^*\|_{\mathcal{H}}^2\Big(\sum_{j>d}\mu_j\Big)\sum_{l=1}^d\mu_\ell.$$

Dividing by n^2, recalling the definition of $\beta_d = \sum_{j>d}\mu_j$, and noting that $\operatorname{tr}(K) \geq \sum_{l=1}^d \mu_\ell$ shows that

$$\mathbb{E}\left[\left\|\frac{1}{n}M^{1/2}\Phi^T v\right\|_2^2\right] \leq \rho^4\|f^*\|_{\mathcal{H}}^2\beta_d\operatorname{tr}(K).$$

Combining this inequality with our expansion (50) and the bound (51) yields the claim (47b).

Proof of bound (48)*:* We consider the expectation of the norm of $Q^{-1}(\frac{1}{n}\Phi^T\Phi - I)Q^{-1}$. For each $i \in [n]$, $\pi_i := (\phi_1(x_i), \ldots, \phi_d(x_i))^T \in \mathbb{R}^d$, then π_i^T is the i-th row of the matrix $\Phi \in \mathbb{R}^{n\times d}$. Then we know that

$$Q^{-1}\left(\frac{1}{n}\Phi^T\Phi - I\right)Q^{-1} = \frac{1}{n}\sum_{i=1}^n Q^{-1}(\pi_i\pi_i^T - I)Q^{-1}.$$

Define the sequence of matrices

$$A_i := Q^{-1}(\pi_i\pi_i^T - I)Q^{-1}$$

Then the matrices $A_i = A_i^T \in \mathbb{R}^{d\times d}$. Note that $\mathbb{E}[A_i] = 0$ and let ε_i be i.i.d. $\{-1, 1\}$-valued Rademacher random variables. Applying a standard symmetrization argument (Ledoux and Talagrand, 1991), we find that for any $k \geq 1$, we have

$$\mathbb{E}\left[\left|\left|\left|Q^{-1}\left(\frac{1}{n}\Phi^T\Phi - I\right)Q^{-1}\right|\right|\right|^k\right] = \mathbb{E}\left[\left|\left|\left|\frac{1}{n}\sum_{i=1}^n A_i\right|\right|\right|^k\right] \leq 2^k\mathbb{E}\left[\left|\left|\left|\frac{1}{n}\sum_{i=1}^n \varepsilon_i A_i\right|\right|\right|^k\right]. \tag{53}$$

Lemma 11 *The quantity* $\mathbb{E}\left[\left|\left|\left|\frac{1}{n}\sum_{i=1}^n \varepsilon_i A_i\right|\right|\right|^k\right]^{1/k}$ *is upper bounded by*

$$\sqrt{e(k \vee 2\log(d))}\frac{\rho^2 \sum_{j=1}^d \frac{1}{1+\lambda/\mu_j}}{\sqrt{n}} + \frac{4e(k \vee 2\log(d))}{n^{1-1/k}}\left(\sum_{j=1}^d \frac{\rho^2}{1+\lambda/\mu_j}\right). \tag{54}$$

We take this lemma as given for the moment, returning to prove it shortly. Recall the definition of the constant $\gamma(\lambda) = \sum_{j=1}^\infty 1/(1+\lambda/\mu_j) \geq \sum_{j=1}^d 1/(1+\lambda/\mu_j)$. Then using our symmetrization inequality (53), we have

$$\begin{aligned}
&\mathbb{E}\left[\left|\left|\left|Q^{-1}\left(\frac{1}{n}\Phi^T\Phi - I\right)Q^{-1}\right|\right|\right|^k\right] \\
&\leq 2^k\left(\sqrt{e(k \vee \log(d))}\frac{\rho^2\gamma(\lambda)}{\sqrt{n}} + \frac{4e(k \vee 2\log(d))}{n^{1-1/k}}\rho^2\gamma(\lambda)\right)^k \\
&\leq \max\left\{\sqrt{k \vee \log(d)}, \frac{k \vee \log(d)}{n^{1/2-1/k}}\right\}^k\left(\frac{C\rho^2\gamma(\lambda)}{\sqrt{n}}\right)^k,
\end{aligned} \tag{55}$$

where C is a numerical constant. By definition of the event $\mathcal{E}$, we see by Markov's inequality that for any $k \in \mathbb{R}, k \geq 1$,

$$\mathbb{P}(\mathcal{E}^c) \leq \frac{\mathbb{E}\left[\left|\left|\left|Q^{-1}\left(\frac{1}{n}\Phi^T\Phi - I\right)\right|\right|\right|^k\right]}{2^{-k}} \leq \max\left\{\sqrt{k \vee \log(d)}, \frac{k \vee \log(d)}{n^{1/2-1/k}}\right\}^k\left(\frac{2C\rho^2\gamma(\lambda))}{\sqrt{n}}\right)^k.$$

This completes the proof of the bound (48).

It remains to prove Lemma 11, for which we make use of the following result, due to Chen et al. (2012, Theorem A.1(2)).

Lemma 12 *Let* $X_i \in \mathbb{R}^{d\times d}$ *be independent symmetrically distributed Hermitian matrices. Then*

$$\mathbb{E}\left[\left|\left|\left|\sum_{i=1}^n X_i\right|\right|\right|^k\right]^{1/k} \leq \sqrt{e(k \vee 2\log d)}\left|\left|\left|\sum_{i=1}^n \mathbb{E}[X_i^2]\right|\right|\right|^{1/2} + 2e(k \vee 2\log d)\left(\mathbb{E}[\max_i |||X_i|||^k]\right)^{1/k}. \tag{56}$$

The proof of Lemma 11 is based on applying this inequality with $X_i = \varepsilon_i A_i/n$, and then bounding the two terms on the right-hand side of inequality (56).

We begin with the first term. Note that for any symmetric matrix Z, we have the matrix inequalities $0 \preceq \mathbb{E}[(Z - \mathbb{E}[Z])^2] = \mathbb{E}[Z^2] - \mathbb{E}[Z]^2 \preceq \mathbb{E}[Z^2]$, so

$$\mathbb{E}[A_i^2] = \mathbb{E}[Q^{-1}(\pi_i\pi_i^T - I)Q^{-2}(\pi_i\pi_i^T - I)Q^{-1}] \preceq \mathbb{E}[Q^{-1}\pi_i\pi_i^T Q^{-2}\pi_i\pi_i^T Q^{-1}].$$

Instead of computing these moments directly, we provide bounds on their norms. Since $\pi_i\pi_i^T$ is rank one and Q is diagonal, we have

$$|||Q^{-1}\pi_i\pi_i^T Q^{-1}||| = \pi_i^T(I + \lambda M^{-1})^{-1}\pi_i = \sum_{j=1}^d \frac{\phi_j(x_i)^2}{1+\lambda/\mu_j}.$$

We also note that, for any $k \in \mathbb{R}, k \geq 1$, convexity implies that

$$\left(\sum_{j=1}^{d} \frac{\phi_j(x_i)^2}{1+\lambda/\mu_j}\right)^k = \left(\frac{\sum_{l=1}^d 1/(1+\lambda/\mu_\ell)}{\sum_{l=1}^d 1/(1+\lambda/\mu_\ell)} \sum_{j=1}^{d} \frac{\phi_j(x_i)^2}{1+\lambda/\mu_j}\right)^k$$
$$\leq \left(\sum_{l=1}^{d} \frac{1}{1+\lambda/\mu_\ell}\right)^k \frac{1}{\sum_{l=1}^d 1/(1+\lambda/\mu_\ell)} \sum_{j=1}^{d} \frac{\phi_j(x_i)^{2k}}{1+\lambda/\mu_j},$$

so if $\mathbb{E}[\phi_j(X_i)^{2k}] \leq \rho^{2k}$, we obtain

$$\mathbb{E}\left[\left(\sum_{j=1}^{d} \frac{\phi_j(x_i)^2}{1+\lambda/\mu_j}\right)^k\right] \leq \left(\sum_{j=1}^{d} \frac{1}{1+\lambda/\mu_j}\right)^k \rho^{2k}. \tag{57}$$

The sub-multiplicativity of matrix norms implies $\left|\!\left|\!\left|(Q^{-1}\pi_i\pi_i^T Q^{-1})^2\right|\!\right|\!\right| \leq \left|\!\left|\!\left|Q^{-1}\pi_i\pi_i^T Q^{-1}\right|\!\right|\!\right|^2$, and consequently we have

$$\mathbb{E}\left[\left|\!\left|\!\left|(Q^{-1}\pi_i\pi_i^T Q^{-1})^2\right|\!\right|\!\right|\right] \leq \mathbb{E}\left[\left(\pi_i^T(I+\lambda M^{-1})^{-1}\pi_i\right)^2\right] \leq \rho^4\left(\sum_{j=1}^{d} \frac{1}{1+\lambda/\mu_j}\right)^2,$$

where the final step follows from inequality (57). Combined with first term on the right-hand side of Lemma 12, we have thus obtained the first term on the right-hand side of expression (54).

We now turn to the second term in expression (54). For real $k \geq 1$, we have

$$\mathbb{E}[\max_i \left|\!\left|\!\left|\varepsilon_i A_i/n\right|\!\right|\!\right|^k] = \frac{1}{n^k}\mathbb{E}[\max_i \left|\!\left|\!\left|A_i\right|\!\right|\!\right|^k] \leq \frac{1}{n^k}\sum_{i=1}^{n}\mathbb{E}[\left|\!\left|\!\left|A_i\right|\!\right|\!\right|^k]$$

Since norms are sub-additive, we find that

$$\left|\!\left|\!\left|A_i\right|\!\right|\!\right|^k \leq 2^{k-1}\left(\sum_{j=1}^{d} \frac{\phi_j(x_i)^2}{1+\lambda/\mu_j}\right)^k + 2^{k-1}\left|\!\left|\!\left|Q^{-2}\right|\!\right|\!\right|^k = 2^{k-1}\left(\sum_{j=1}^{d} \frac{\phi_j(x_i)^2}{1+\lambda/\mu_j}\right)^k + 2^{k-1}\left(\frac{1}{1+\lambda/\mu_1}\right)^k.$$

Since $\rho \geq 1$ (recall that the ϕ_j are an orthonormal basis), we apply inequality (57), to find that

$$\mathbb{E}[\max_i \left|\!\left|\!\left|\varepsilon_i A_i/n\right|\!\right|\!\right|^k] \leq \frac{1}{n^{k-1}}\left[2^{k-1}\left(\sum_{j=1}^{d} \frac{1}{1+\lambda/\mu_j}\right)^k \rho^{2k} + 2^{k-1}\left(\frac{1}{1+\lambda/\mu_1}\right)^k \rho^{2k}\right].$$

Taking kth roots yields the second term in the expression (54).

Appendix B. Proof of Lemma 7

This proof follows an outline similar to Lemma 6. We begin with a simple bound on $\|\Delta\|_{\mathcal{H}}$:

Lemma 13 *Under Assumption B, we have* $\mathbb{E}[\|\Delta\|_{\mathcal{H}}^2 \mid X] \leq 2\sigma^2/\lambda + 4\|f^*\|_{\mathcal{H}}^2$.

Proof We have

$$\begin{aligned}\lambda\,\mathbb{E}[\,\|\widehat{f}\|_{\mathcal{H}}^2 \mid \{x_i\}_{i=1}^n] &\leq \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^n (\widehat{f}(x_i) - f^*(x_i) - \varepsilon_i)^2 + \lambda\|\widehat{f}\|_{\mathcal{H}}^2 \mid \{x_i\}_{i=1}^n\right] \\ &\overset{(i)}{\leq} \frac{1}{n}\sum_{i=1}^n \mathbb{E}[\varepsilon_i^2 \mid x_i] + \lambda\,\|f^*\|_{\mathcal{H}}^2 \\ &\overset{(ii)}{\leq} \sigma^2 + \lambda\,\|f^*\|_{\mathcal{H}}^2\,,\end{aligned}$$

where inequality (i) follows since $\widehat{f}$ minimizes the objective function (2); and inequality (ii) uses the fact that $\mathbb{E}[\varepsilon_i^2 \mid x_i] \leq \sigma^2$. Applying the triangle inequality to $\|\Delta\|_{\mathcal{H}}$ along with the elementary inequality $(a+b)^2 \leq 2a^2 + 2b^2$, we find that

$$\mathbb{E}[\|\Delta\|_{\mathcal{H}}^2 \mid \{x_i\}_{i=1}^n] \leq 2\,\|f^*\|_{\mathcal{H}}^2 + 2\mathbb{E}[\|\widehat{f}\|_{\mathcal{H}}^2 \mid \{x_i\}_{i=1}^n] \;\leq\; \frac{2\sigma^2}{\lambda} + 4\,\|f^*\|_{\mathcal{H}}^2\,,$$

which completes the proof. ■

With Lemma 13 in place, we now proceed to the proof of the theorem proper. Recall from Lemma 6 the optimality condition

$$\frac{1}{n}\sum_{i=1}^n \xi_{x_i}(\langle \xi_{x_i}, \widehat{f} - f^*\rangle - \varepsilon_i) + \lambda\widehat{f} = 0. \tag{58}$$

Now, let $\delta \in \ell_2(\mathbb{N})$ be the expansion of the error Δ in the basis $\{\phi_j\}$, so that $\Delta = \sum_{j=1}^\infty \delta_j\phi_j$, and (again, as in Lemma 6), we choose $d \in \mathbb{N}$ and truncate Δ via

$$\Delta^\downarrow := \sum_{j=1}^d \delta_j\phi_j \;\text{ and }\; \Delta^\uparrow := \Delta - \Delta^\downarrow = \sum_{j>d}\delta_j\phi_j.$$

Let $\delta^\downarrow \in \mathbb{R}^d$ and $\delta^\uparrow$ denote the corresponding vectors for the above. As a consequence of the orthonormality of the basis functions, we have

$$\mathbb{E}[\|\Delta\|_2^2] = \mathbb{E}[\|\Delta^\downarrow\|_2^2] + \mathbb{E}[\|\Delta^\uparrow\|_2^2] = \mathbb{E}[\|\delta^\downarrow\|_2^2] + \mathbb{E}[\|\delta^\uparrow\|_2^2]. \tag{59}$$

We bound each of the terms (59) in turn.

By Lemma 13, the second term is upper bounded as

$$\mathbb{E}[\|\Delta^\uparrow\|_2^2] = \sum_{j>d}\mathbb{E}[\delta_j^2] \leq \sum_{j>d}\frac{\mu_{d+1}}{\mu_j}\mathbb{E}[\delta_j^2] = \mu_{d+1}\mathbb{E}[\|\Delta^\uparrow\|_{\mathcal{H}}^2] \leq \mu_{d+1}\left(\frac{2\sigma^2}{\lambda} + 4\,\|f^*\|_{\mathcal{H}}^2\right). \tag{60}$$

The remainder of the proof is devoted the bounding the term $\mathbb{E}[\|\Delta^\downarrow\|_2^2]$ in the decomposition (59). By taking the Hilbert inner product of ϕ_k with the optimality condition (58), we find as in our derivation of the matrix equation (45) that for each $k \in \{1,\ldots,d\}$

$$\frac{1}{n}\sum_{i=1}^n\sum_{j=1}^d \phi_k(x_i)\phi_j(x_i)\delta_j + \frac{1}{n}\sum_{i=1}^n \phi_k(x_i)(\Delta^\uparrow(x_i) - \varepsilon_i) + \lambda\frac{\delta_k}{\mu_k} = 0.$$

Given the expansion $f^* = \sum_{j=1}^{\infty} \theta_j \phi_j$, define the tail error vector $v \in \mathbb{R}^n$ by $v_i = \sum_{j>d} \delta_j \phi_j(x_i)$, and recall the definition of the eigenvalue matrix $M = \operatorname{diag}(\mu_1, \ldots, \mu_d) \in \mathbb{R}^{d \times d}$. Given the matrix Φ defined by its coordinates $\Phi_{ij} = \phi_j(x_i)$, we have

$$\left(\frac{1}{n}\Phi^T\Phi + \lambda M^{-1}\right)\delta^{\downarrow} = -\lambda M^{-1}\theta^{\downarrow} - \frac{1}{n}\Phi^T v + \frac{1}{n}\Phi^T\varepsilon. \tag{61}$$

As in the proof of Lemma 6, we find that

$$\left(I + Q^{-1}\left(\frac{1}{n}\Phi^T\Phi - I\right)Q^{-1}\right)Q\delta^{\downarrow} = -\lambda Q^{-1}M^{-1}\theta^{\downarrow} - \frac{1}{n}Q^{-1}\Phi^T v + \frac{1}{n}Q^{-1}\Phi^T\varepsilon, \tag{62}$$

where we recall that $Q = (I + \lambda M^{-1})^{1/2}$.

We now recall the bounds (47a) and (48) from Lemma 10, as well as the previously defined event $\mathcal{E} := \{\left|\left|\left|Q^{-1}\left(\frac{1}{n}\Phi^T\Phi - I\right)Q^{-1}\right|\right|\right| \leq 1/2\}$. When $\mathcal{E}$ occurs, the expression (62) implies the inequality

$$\|\Delta^{\downarrow}\|_2^2 \leq \|Q\delta^{\downarrow}\|_2^2 \leq 4\left\|-\lambda Q^{-1}M^{-1}\theta^{\downarrow} - (1/n)Q^{-1}\Phi^T v + (1/n)Q^{-1}\Phi^T\varepsilon\right\|_2^2.$$

When $\mathcal{E}$ fails to hold, Lemma 13 may still be applied since $\mathcal{E}$ is measurable with respect to $\{x_i\}_{i=1}^n$. Doing so yields

$$\begin{aligned}
&\mathbb{E}[\|\Delta^{\downarrow}\|_2^2] = \mathbb{E}[1(\mathcal{E})\,\|\Delta^{\downarrow}\|_2^2] + \mathbb{E}[1(\mathcal{E}^c)\,\|\Delta^{\downarrow}\|_2^2] \\
&\leq 4\mathbb{E}\left[\left\|-\lambda Q^{-1}M^{-1}\theta^{\downarrow} - (1/n)Q^{-1}\Phi^T v + (1/n)Q^{-1}\Phi^T\varepsilon\right\|_2^2\right] + \mathbb{E}\left[1(\mathcal{E}^c)\,\mathbb{E}[\|\Delta^{\downarrow}\|_2^2 \mid \{x_i\}_{i=1}^n]\right] \\
&\leq 4\mathbb{E}\left[\left\|\lambda Q^{-1}M^{-1}\theta^{\downarrow} + \frac{1}{n}Q^{-1}\Phi^T v - \frac{1}{n}Q^{-1}\Phi^T\varepsilon\right\|_2^2\right] + \mathbb{P}(\mathcal{E}^c)\left(\frac{2\sigma^2}{\lambda} + 4\,\|f^*\|_{\mathcal{H}}^2\right).
\end{aligned} \tag{63}$$

Since the bound (48) still holds, it remains to provide a bound on the first term in the expression (63).

As in the proof of Lemma 6, we have $\|\lambda Q^{-1}M^{-1}\theta^{\downarrow}\|_2^2 \leq \lambda\,\|f^*\|_{\mathcal{H}}^2$ via the bound (47a). Turning to the second term inside the norm, we claim that, under the conditions of Lemma 7, the following bound holds:

$$\mathbb{E}\left[\left\|(1/n)Q^{-1}\Phi^T v\right\|_2^2\right] \leq \frac{\rho^4\,\operatorname{tr}(K)\beta_d(2\sigma^2/\lambda + 4\,\|f^*\|_{\mathcal{H}}^2)}{\lambda}. \tag{64}$$

This claim is an analogue of our earlier bound (47b), and we prove it shortly. Lastly, we bound the norm of $Q^{-1}\Phi^T\varepsilon/n$. Noting that the diagonal entries of Q^{-1} are $1/\sqrt{1+\lambda/\mu_j}$, we have

$$\mathbb{E}\left[\left\|Q^{-1}\Phi^T\varepsilon\right\|_2^2\right] = \sum_{j=1}^{d}\sum_{i=1}^{n}\frac{1}{1+\lambda/\mu_j}\mathbb{E}[\phi_j^2(X_i)\varepsilon_i^2]$$

Since $\mathbb{E}[\phi_j^2(X_i)\varepsilon_i^2] = \mathbb{E}[\phi_j^2(X_i)\mathbb{E}[\varepsilon_i^2 \mid X_i]] \leq \sigma^2$ by assumption, we have the inequality

$$\mathbb{E}\left[\left\|(1/n)Q^{-1}\Phi^T\varepsilon\right\|_2^2\right] \leq \frac{\sigma^2}{n}\sum_{j=1}^{d}\frac{1}{1+\lambda/\mu_j}.$$

The last sum is bounded by $(\sigma^2/n)\gamma(\lambda)$. Applying the inequality $(a+b+c)^2 \leq 3a^2+3b^2+3c^2$ to inequality (63), we obtain

$$\mathbb{E}\left[\|\Delta^{\downarrow}\|_2^2\right] \leq 12\lambda \|f^*\|_{\mathcal{H}}^2 + \frac{12\sigma^2\gamma(\lambda)}{n} + \left(\frac{2\sigma^2}{\lambda} + 4\|f^*\|_{\mathcal{H}}^2\right)\left(\frac{12\rho^4 \operatorname{tr}(K)\beta_d}{\lambda} + \mathbb{P}(\mathcal{E}^c)\right).$$

Applying the bound (48) to control $\mathbb{P}(\mathcal{E}^c)$ and bounding $\mathbb{E}[\|\Delta^{\uparrow}\|_2^2]$ using inequality (60) completes the proof of the lemma.

It remains to prove bound (64). Recalling the inequality (51), we see that

$$\left\|(1/n)Q^{-1}\Phi^T v\right\|_2^2 \leq \left|\!\left|\!\left|Q^{-1}M^{-1/2}\right|\!\right|\!\right|^2 \left\|(1/n)M^{1/2}\Phi^T v\right\|_2^2 \leq \frac{1}{\lambda}\left\|(1/n)M^{1/2}\Phi^T v\right\|_2^2. \quad (65)$$

Let Φ_ℓ denote the ℓth column of the matrix Φ. Taking expectations yields

$$\mathbb{E}\left[\left\|M^{1/2}\Phi^T v\right\|_2^2\right] = \sum_{l=1}^d \mu_\ell \mathbb{E}[\langle \Phi_\ell, v\rangle^2] \leq \sum_{l=1}^d \mu_\ell \mathbb{E}\left[\|\Phi_\ell\|_2^2 \|v\|_2^2\right] = \sum_{l=1}^d \mu_\ell \mathbb{E}\left[\|\Phi_\ell\|_2^2 \,\mathbb{E}\left[\|v\|_2^2 \mid X\right]\right].$$

Now consider the inner expectation. Applying the Cauchy-Schwarz inequality as in the proof of the bound (47b), we have

$$\|v\|_2^2 = \sum_{i=1}^n v_i^2 \leq \sum_{i=1}^n \left(\sum_{j>d} \frac{\delta_j^2}{\mu_j}\right)\left(\sum_{j>d} \mu_j \phi_j^2(X_i)\right).$$

Notably, the second term is X-measurable, and the first is bounded by $\|\Delta^{\uparrow}\|_{\mathcal{H}}^2 \leq \|\Delta\|_{\mathcal{H}}^2$. We thus obtain

$$\mathbb{E}\left[\left\|M^{1/2}\Phi^T v\right\|_2^2\right] \leq \sum_{i=1}^n \sum_{l=1}^d \mu_\ell \mathbb{E}\left[\|\Phi_\ell\|_2^2 \left(\sum_{j>d} \mu_j \phi_j^2(X_i)\right)\mathbb{E}[\|\Delta\|_{\mathcal{H}}^2 \mid X]\right]. \quad (66)$$

Lemma 13 provides the bound $2\sigma^2/\lambda + 4\|f^*\|_{\mathcal{H}}^2$ on the final (inner) expectation.

The remainder of the argument proceeds precisely as in the bound (47b). We have

$$\mathbb{E}[\|\Phi_\ell\|_2^2 \,\phi_j(X_i)^2] \leq n\rho^4$$

by the moment assumptions on ϕ_j, and thus

$$\mathbb{E}\left[\left\|M^{1/2}\Phi^T v\right\|_2^2\right] \leq \sum_{l=1}^d \sum_{j>d} \mu_\ell \mu_j n^2 \rho^4 \left(\frac{2\sigma^2}{\lambda} + 4\|f^*\|_{\mathcal{H}}^2\right) \leq n^2\rho^4 \beta_d \operatorname{tr}(K)\left(\frac{2\sigma^2}{\lambda} + 4\|f^*\|_{\mathcal{H}}^2\right).$$

Dividing by λn^2 completes the proof.

Appendix C. Proof of Lemma 8

As before, we let $\{x_i\}_{i=1}^n := \{x_1, \ldots, x_n\}$ denote the collection of design points. We begin with some useful bounds on $\left\|f_{\bar{\lambda}}^*\right\|_{\mathcal{H}}$ and $\|\Delta\|_{\mathcal{H}}$.

Lemma 14 *Under Assumptions A and B′, we have*

$$\mathbb{E}\left[(\mathbb{E}[\|\Delta\|_{\mathcal{H}}^2 \mid \{x_i\}_{i=1}^n])^2\right] \le B_{\lambda,\bar{\lambda}}^4 \quad \text{and} \quad \mathbb{E}[\|\Delta\|_{\mathcal{H}}^2] \le B_{\lambda,\bar{\lambda}}^2, \tag{67}$$

where

$$B_{\lambda,\bar{\lambda}} := \sqrt[4]{32\|f_{\bar{\lambda}}^*\|_{\mathcal{H}}^4 + 8\tau_{\bar{\lambda}}^4/\lambda^2}. \tag{68}$$

See Section C.1 for the proof of this claim.

This proof follows an outline similar to that of Lemma 7. As usual, we let $\delta \in \ell_2(\mathbb{N})$ be the expansion of the error Δ in the basis $\{\phi_j\}$, so that $\Delta = \sum_{j=1}^\infty \delta_j \phi_j$, and we choose $d \in \mathbb{N}$ and define the truncated vectors $\Delta^\downarrow := \sum_{j=1}^d \delta_j \phi_j$ and $\Delta^\uparrow := \Delta - \Delta^\downarrow = \sum_{j>d} \delta_j \phi_j$. As usual, we have the decomposition $\mathbb{E}[\|\Delta\|_2^2] = \mathbb{E}[\|\delta^\downarrow\|_2^2] + \mathbb{E}[\|\delta^\uparrow\|_2^2]$. Recall the definition (68) of the constant $B_{\lambda,\bar{\lambda}} = \sqrt[4]{32\|f_{\bar{\lambda}}^*\|_{\mathcal{H}}^4 + 8\tau_{\bar{\lambda}}^4/\lambda^2}$. As in our deduction of inequalities (60), Lemma 14 implies that $\mathbb{E}[\|\Delta^\uparrow\|_2^2] \le \mu_{d+1}\mathbb{E}[\|\Delta^\uparrow\|_{\mathcal{H}}^2] \le \mu_{d+1} B_{\lambda,\bar{\lambda}}^2$.

The remainder of the proof is devoted to bounding $\mathbb{E}[\|\delta^\downarrow\|_2^2]$. We use identical notation to that in our proof of Lemma 7, which we recap for reference (see also Table 2). We define the tail error vector $v \in \mathbb{R}^n$ by $v_i = \sum_{j>d} \delta_j \phi_j(x_i)$, $i \in [n]$, and recall the definitions of the eigenvalue matrix $M = \operatorname{diag}(\mu_1, \ldots, \mu_d) \in \mathbb{R}^{d \times d}$ and basis matrix Φ with $\Phi_{ij} = \phi_j(x_i)$. We use $Q = (I + \lambda M^{-1})^{1/2}$ for shorthand, and we let $\mathcal{E}$ be the event that

$$\left\|\left\|Q^{-1}((1/n)\Phi^T\Phi - I)Q^{-1}\right\|\right\| \le 1/2.$$

Writing $f_{\bar{\lambda}}^* = \sum_{j=1}^\infty \theta_j \phi_j$, we define the alternate noise vector $\varepsilon_i' = Y_i - f_{\bar{\lambda}}^*(x_i)$. Using this notation, mirroring the proof of Lemma 7 yields

$$\mathbb{E}[\|\Delta^\downarrow\|_2^2] \le \mathbb{E}[\|Q\delta^\downarrow\|_2^2] \le 4\mathbb{E}\left[\left\|\lambda Q^{-1}M^{-1}\theta^\downarrow + \frac{1}{n}Q^{-1}\Phi^T v - \frac{1}{n}Q^{-1}\Phi^T\varepsilon'\right\|_2^2\right] + \mathbb{P}(\mathcal{E}^c)B_{\lambda,\bar{\lambda}}^2, \tag{69}$$

which is an analogue of equation (63). The bound bound (48) controls the probability $\mathbb{P}(\mathcal{E}^c)$, so it remains to control the first term in the expression (69). We first rewrite the expression within the norm as

$$(\lambda - \bar{\lambda})Q^{-1}M^{-1}\theta^\downarrow + \frac{1}{n}Q^{-1}\Phi^T v - \left(\frac{1}{n}Q^{-1}\Phi^T\varepsilon' - \bar{\lambda}Q^{-1}M^{-1}\theta^\downarrow\right)$$

The following lemma provides bounds on the first two terms:

Lemma 15 *The following bounds hold:*

$$\left\|(\bar{\lambda} - \lambda)Q^{-1}M^{-1}\theta^\downarrow\right\|_2^2 \le \frac{(\bar{\lambda} - \lambda)^2 \left\|f_{\bar{\lambda}}^*\right\|_{\mathcal{H}}^2}{\lambda}, \tag{70a}$$

$$\mathbb{E}\left[\left\|\frac{1}{n}Q^{-1}\Phi^T v\right\|_2^2\right] \le \frac{\rho^4 B_{\lambda,\bar{\lambda}}^2 \operatorname{tr}(K)\beta_d}{\lambda}, \tag{70b}$$

For the third term, we make the following claim.

Lemma 16 *Under Assumptions A and B′, we have*

$$\mathbb{E}\left[\left\|\frac{1}{n}Q^{-1}\Phi^T\varepsilon' - \bar{\lambda}Q^{-1}M^{-1}\theta^{\downarrow}\right\|_2^2\right] \leq \frac{\gamma(\lambda)\rho^2\tau_{\bar{\lambda}}^2}{n}. \tag{71}$$

Deferring the proof of the two lemmas to Sections C.2 and C.3, we apply the inequality $(a+b+c)^2 \leq 4a^2 + 4b^2 + 2c^2$ to inequality (69), and we have

$$\begin{aligned}
&\mathbb{E}[\|\Delta^{\downarrow}\|_2^2] - \mathbb{P}(\mathcal{E}^c)B_{\lambda,\bar{\lambda}}^2 \leq \mathbb{E}[\|Q\delta^{\downarrow}\|_2^2] - \mathbb{P}(\mathcal{E}^c)B_{\lambda,\bar{\lambda}}^2 \\
&\leq 16\mathbb{E}\left[\left\|(\lambda-\bar{\lambda})Q^{-1}M^{-1}\theta^{\downarrow}\right\|_2^2\right] + \frac{16}{n^2}\mathbb{E}\left[\left\|Q^{-1}\Phi^T v\right\|_2^2\right] + \frac{8}{n^2}\mathbb{E}\left[\left\|Q^{-1}\Phi^T\varepsilon' - \bar{\lambda}Q^{-1}M^{-1}\theta^{\downarrow}\right\|_2^2\right] \\
&\leq \frac{16(\bar{\lambda}-\lambda)^2\left\|f_{\bar{\lambda}}^*\right\|_{\mathcal{H}}^2}{\lambda} + \frac{16\rho^4 B_{\lambda,\bar{\lambda}}^2\operatorname{tr}(K)\beta_d}{\lambda} + \frac{8\gamma(\lambda)\rho^2\tau_{\bar{\lambda}}^2}{n},
\end{aligned} \tag{72}$$

where we have applied the bounds (70a) and (70b) from Lemma 17 and the bound (71) from Lemma 16. Applying the bound (48) to control $\mathbb{P}(\mathcal{E}^c)$ and recalling that $\mathbb{E}[\left\|\Delta^{\uparrow}\right\|_2^2] \leq \mu_{d+1}B_{\lambda,\bar{\lambda}}^2$ completes the proof.

C.1 Proof of Lemma 14

Recall that $\widehat{f}$ minimizes the empirical objective. Consequently,

$$\begin{aligned}
\lambda\mathbb{E}[\|\widehat{f}\|_{\mathcal{H}}^2 \mid \{x_i\}_{i=1}^n] &\leq \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^n(\widehat{f}(x_i)-Y_i)^2 + \lambda\|\widehat{f}\|_{\mathcal{H}}^2 \mid \{x_i\}_{i=1}^n\right] \\
&\leq \frac{1}{n}\sum_{i=1}^n\mathbb{E}[(f_{\bar{\lambda}}^*(x_i)-Y_i)^2 \mid x_i] + \lambda\|f_{\bar{\lambda}}^*\|_{\mathcal{H}}^2 = \frac{1}{n}\sum_{i=1}^n\sigma_{\bar{\lambda}}^2(x_i) + \lambda\|f_{\bar{\lambda}}^*\|_{\mathcal{H}}^2
\end{aligned}$$

The triangle inequality immediately gives us the upper bound

$$\mathbb{E}[\|\Delta\|_{\mathcal{H}}^2 \mid \{x_i\}_{i=1}^n] \leq 2\|f_{\bar{\lambda}}^*\|_{\mathcal{H}}^2 + \mathbb{E}[2\|\widehat{f}\|_{\mathcal{H}}^2 \mid \{x_i\}_{i=1}^n] \leq \frac{2}{\lambda n}\sum_{i=1}^n\sigma_{\bar{\lambda}}^2(x_i) + 4\|f_{\bar{\lambda}}^*\|_{\mathcal{H}}^2.$$

Since $(a+b)^2 \leq 2a^2 + 2b^2$, convexity yields

$$\begin{aligned}
\mathbb{E}[(\mathbb{E}[\|\Delta\|_{\mathcal{H}}^2 \mid \{x_i\}_{i=1}^n])^2] &\leq \mathbb{E}\left[\left(\frac{2}{\lambda n}\sum_{i=1}^n\sigma_{\bar{\lambda}}^2(X_i) + 4\|f_{\bar{\lambda}}^*\|_{\mathcal{H}}^2\right)^2\right] \\
&\leq \frac{8}{\lambda^2 n}\sum_{i=1}^n\mathbb{E}[\sigma_{\bar{\lambda}}^4(X_i)] + 32\|f_{\bar{\lambda}}^*\|_{\mathcal{H}}^4 = 32\|f_{\bar{\lambda}}^*\|_{\mathcal{H}}^4 + \frac{8\tau_{\bar{\lambda}}^4}{\lambda^2}.
\end{aligned}$$

This completes the proof of the first of the inequalities (67). The second of the inequalities (67) follows from the first by Jensen's inequality.

C.2 Proof of Lemma 15

Our previous bound (47a) immediately implies inequality (70a). To prove the second upper bound, we follow the proof of the bound (64). From inequalities (65) and (66), we obtain that

$$\left\|(1/n)Q^{-1}\Phi^T v\right\|_2^2 \leq \frac{1}{\lambda n^2}\sum_{i=1}^{n}\sum_{l=1}^{d}\sum_{j>d}\mu_\ell\mu_j\mathbb{E}\left[\|\Phi_\ell\|_2^2\,\phi_j^2(X_i)\mathbb{E}[\|\Delta\|_{\mathcal{H}}^2 \mid \{X_i\}_{i=1}^n]\right]. \tag{73}$$

Applying Hölder's inequality yields

$$\mathbb{E}\left[\|\Phi_\ell\|_2^2\,\phi_j^2(X_i)\mathbb{E}[\|\Delta\|_{\mathcal{H}}^2 \mid \{X_i\}_{i=1}^n]\right] \leq \sqrt{\mathbb{E}[\|\Phi_\ell\|_2^4\,\phi_j^4(X_i)]}\sqrt{\mathbb{E}[(\mathbb{E}[\|\Delta\|_{\mathcal{H}}^2 \mid \{X_i\}_{i=1}^n])^2]}.$$

Note that Lemma 14 provides the bound $B_{\lambda,\bar{\lambda}}^4$ on the final expectation. By definition of Φ_ℓ, we find that

$$\mathbb{E}[\|\Phi_\ell\|_2^4\,\phi_j^4(x_i)] = \mathbb{E}\left[\left(\sum_{k=1}^{n}\phi_\ell^2(x_k)\right)^2\phi_j^4(x_i)\right] \leq n^2\mathbb{E}\left[\frac{1}{2}\left(\phi_\ell^8(x_1)+\phi_j^8(x_1)\right)\right] \leq n^2\rho^8,$$

where we have used Assumption A with moment $2k \geq 8$, or equivalently $k \geq 4$. Thus

$$\mathbb{E}\left[\|\Phi_\ell\|_2^2\,\phi_j^2(X_i)\mathbb{E}[\|\Delta\|_{\mathcal{H}}^2 \mid \{X_i\}_{i=1}^n]\right] \leq n\rho^4 B_{\lambda,\bar{\lambda}}^2. \tag{74}$$

Combining inequalities (73) and (74) yields the bound (70b).

C.3 Proof of Lemma 16

Using the fact that Q and M are diagonal, we have

$$\mathbb{E}\left[\left\|\frac{1}{n}Q^{-1}\Phi^T\varepsilon' - \bar{\lambda}Q^{-1}M^{-1}\theta^{\downarrow}\right\|_2^2\right] = \sum_{j=1}^{d}Q_{jj}^{-2}\mathbb{E}\left[\left(\frac{1}{n}\sum_{i=1}^{n}\phi_j(X_i)\varepsilon_i' - \frac{\bar{\lambda}\theta_j}{\mu_j}\right)^2\right]. \tag{75}$$

Fréchet differentiability and the fact that $f_{\bar{\lambda}}^*$ is the global minimizer of the regularized regression problem imply that

$$\mathbb{E}[\xi_{X_i}\varepsilon_i'] + \bar{\lambda}f_{\bar{\lambda}}^* = \mathbb{E}\left[\xi_X\left(\langle\xi_X, f_{\bar{\lambda}}^*\rangle - y\right)\right] + \bar{\lambda}f_{\bar{\lambda}}^* = 0.$$

Taking the (Hilbert) inner product of the preceding display with the basis function ϕ_j, we get

$$\mathbb{E}\left[\phi_j(X_i)\varepsilon_i' - \frac{\bar{\lambda}\theta_j}{\mu_j}\right] = 0. \tag{76}$$

Combining the equalities (75) and (76) and using the i.i.d. nature of $\{x_i\}_{i=1}^n$ leads to

$$\begin{aligned}\mathbb{E}\left[\left\|\frac{1}{n}Q^{-1}\Phi^T\varepsilon' - \bar{\lambda}Q^{-1}M^{-1}\theta^{\downarrow}\right\|_2^2\right] &= \sum_{j=1}^{d}Q_{jj}^{-2}\operatorname{var}\left(\frac{1}{n}\sum_{i=1}^{n}\phi_j(X_i)\varepsilon_i' - \frac{\bar{\lambda}\theta_j}{\mu_j}\right)\\ &= \frac{1}{n}\sum_{j=1}^{d}Q_{jj}^{-2}\operatorname{var}\left(\phi_j(X_1)\varepsilon_1'\right).\end{aligned} \tag{77}$$

Using the elementary inequality $\mathrm{var}(Z) \leq \mathbb{E}[Z^2]$ for any random variable Z, we have from Hölder's inequality that

$$\mathrm{var}(\phi_j(X_1)\varepsilon_1') \leq \mathbb{E}[\phi_j(X_1)^2(\varepsilon_1')^2] \leq \sqrt{\mathbb{E}[\phi_j(X_1)^4]\mathbb{E}[\sigma_{\bar{\lambda}}^4(X_1)]} \leq \sqrt{\rho^4}\sqrt{\tau_{\bar{\lambda}}^4},$$

where we used Assumption B$'$ to upper bound the fourth moment $\mathbb{E}[\sigma_{\bar{\lambda}}^4(X_1)]$. Using the fact that $Q_{jj}^{-1} \leq 1$, we obtain the following upper bound on the quantity (77):

$$\frac{1}{n}\sum_{j=1}^{d} Q_{jj}^{-2}\,\mathrm{var}(\phi_j(X_1)\varepsilon_1') = \frac{1}{n}\sum_{j=1}^{d}\frac{\mathrm{var}(\phi_j(X_1)\varepsilon_1')}{1+\lambda/\mu_j} \leq \frac{\gamma(\lambda)\rho^2\tau_{\bar{\lambda}}^2}{n},$$

which establishes the claim.

Appendix D. Proof of Lemma 9

At a high-level, the proof is similar to that of Lemma 6, but we take care since the errors $f_{\bar{\lambda}}^*(x) - y$ are not conditionally mean-zero (or of conditionally bounded variance). Recalling our notation of ξ_x as the RKHS evaluator for x, we have by assumption that $\widehat{f}$ minimizes the empirical objective (39). As in our derivation of equality (40), the Fréchet differentiability of this objective implies the first-order optimality condition

$$\frac{1}{n}\sum_{i=1}^{n}\xi_{x_i}\left\langle\xi_{x_i},\Delta\right\rangle + \frac{1}{n}\sum_{i=1}^{n}(\xi_{x_i}\left\langle\xi_{x_i},f_{\bar{\lambda}}^*\right\rangle - y_i) + \lambda\Delta + \lambda f_{\bar{\lambda}}^* = 0, \tag{78}$$

where $\Delta := \widehat{f} - f_{\bar{\lambda}}^*$. In addition, the optimality of $f_{\bar{\lambda}}^*$ implies that $\mathbb{E}[\xi_{x_i}(\langle\xi_{x_i}, f_{\bar{\lambda}}^*\rangle - y_i)] + \bar{\lambda}f_{\bar{\lambda}}^* = 0$. Using this in equality (78), we take expectations with respect to $\{x_i, y_i\}$ to obtain

$$\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}\xi_{X_i}\left\langle\xi_{X_i},\Delta\right\rangle + \lambda\Delta\right] + (\lambda-\bar{\lambda})f_{\bar{\lambda}}^* = 0.$$

Recalling the definition of the sample covariance operator $\widehat{\Sigma} := \frac{1}{n}\sum_{i=1}^{n}\xi_{x_i}\otimes\xi_{x_i}$, we arrive at

$$\mathbb{E}[(\widehat{\Sigma}+\lambda I)\Delta] = (\bar{\lambda}-\lambda)f_{\bar{\lambda}}^*, \tag{79}$$

which is the analogue of our earlier equality (41).

We now proceed via a truncation argument similar to that used in our proofs of Lemmas 6 and 7. Let $\delta \in \ell_2(\mathbb{N})$ be the expansion of the error Δ in the basis $\{\phi_j\}$, so that $\Delta = \sum_{j=1}^{\infty}\delta_j\phi_j$. For a fixed (arbitrary) $d \in \mathbb{N}$, define

$$\Delta^{\downarrow} := \sum_{j=1}^{d}\delta_j\phi_j \quad \text{and} \quad \Delta^{\uparrow} := \Delta - \Delta^{\downarrow} = \sum_{j>d}\delta_j\phi_j,$$

and note that $\|\mathbb{E}[\Delta]\|_2^2 = \|\mathbb{E}[\Delta^{\downarrow}]\|_2^2 + \|\mathbb{E}[\Delta^{\uparrow}]\|_2^2$. By Lemma 14, the second term is controlled by

$$\|E[\Delta^{\uparrow}]\|_2^2 \leq \mathbb{E}[\|\Delta^{\uparrow}\|_2^2] = \sum_{j>d}\mathbb{E}[\delta_j^2] \leq \sum_{j>d}\frac{\mu_{d+1}}{\mu_j}\mathbb{E}[\delta_j^2] = \mu_{d+1}\mathbb{E}[\|\Delta^{\uparrow}\|_{\mathcal{H}}^2] \leq \mu_{d+1}B_{\lambda,\bar{\lambda}}^2. \tag{80}$$

The remainder of the proof is devoted to bounding $\|\mathbb{E}[\Delta^{\downarrow}]\|_2^2$. Let $f^*_{\bar{\lambda}}$ have the expansion $(\theta_1, \theta_2, \ldots)$ in the basis $\{\phi_j\}$. Recall (as in Lemmas 6 and 7) the definition of the matrix $\Phi \in \mathbb{R}^{n \times d}$ by its coordinates $\Phi_{ij} = \phi_j(x_i)$, the diagonal matrix $M = \operatorname{diag}(\mu_1, \ldots, \mu_d) \succ 0 \in \mathbb{R}^{d \times d}$, and the tail error vector $v \in \mathbb{R}^n$ by $v_i = \sum_{j>d} \delta_j \phi_j(x_i) = \Delta^{\uparrow}(x_i)$. Proceeding precisely as in the derivations of equalities (45) and (61), we have the following equality:

$$\mathbb{E}\left[\left(\frac{1}{n}\Phi^T\Phi + \lambda M^{-1}\right)\delta^{\downarrow}\right] = (\bar{\lambda} - \lambda)M^{-1}\theta^{\downarrow} - \mathbb{E}\left[\frac{1}{n}\Phi^T v\right]. \tag{81}$$

Recalling the definition of the shorthand matrix $Q = (I + \lambda M^{-1})^{1/2}$, with some algebra we have

$$Q^{-1}\left(\frac{1}{n}\Phi^T\Phi + \lambda M^{-1}\right) = Q + Q^{-1}\left(\frac{1}{n}\Phi^T\Phi - I\right),$$

so we can expand expression (81) as

$$\begin{aligned}\mathbb{E}\left[Q\delta^{\downarrow} + Q^{-1}\left(\frac{1}{n}\Phi\Phi^T - I\right)\delta^{\downarrow}\right] &= \mathbb{E}\left[Q^{-1}\left(\frac{1}{n}\Phi^T\Phi + \lambda M^{-1}\right)\delta^{\downarrow}\right] \\ &= (\bar{\lambda} - \lambda)Q^{-1}M^{-1}\theta^{\downarrow} - \mathbb{E}\left[\frac{1}{n}Q^{-1}\Phi^T v\right],\end{aligned}$$

or, rewriting,

$$\mathbb{E}[Q\delta^{\downarrow}] = (\bar{\lambda} - \lambda)Q^{-1}M^{-1}\theta^{\downarrow} - \mathbb{E}\left[\frac{1}{n}Q^{-1}\Phi^T v\right] - \mathbb{E}\left[Q^{-1}\left(\frac{1}{n}\Phi^T\Phi - I\right)\delta^{\downarrow}\right]. \tag{82}$$

Lemma 15 provides bounds on the first two terms on the right-hand-side of equation (82). The following lemma provides upper bounds on the third term:

Lemma 17 *There exists a universal constant C such that*

$$\left\|\mathbb{E}\left[Q^{-1}\left(\frac{1}{n}\Phi^T\Phi - I\right)\delta^{\downarrow}\right]\right\|_2^2 \leq \frac{C(\rho^2\gamma(\lambda)\log d)^2}{n}\mathbb{E}\left[\|Q\delta^{\downarrow}\|_2^2\right], \tag{83}$$

We defer the proof to Section D.1.

Applying Lemma 15 and Lemma 17 to equality (82) and using the standard inequality $(a+b+c)^2 \leq 4a^2 + 4b^2 + 2c^2$, we obtain the upper bound

$$\left\|\mathbb{E}[\Delta^{\downarrow}]\right\|_2^2 \leq \frac{4(\bar{\lambda} - \lambda)^2\left\|f^*_{\bar{\lambda}}\right\|_{\mathcal{H}}^2}{\lambda} + \frac{4\rho^4 B^2_{\lambda,\bar{\lambda}}\operatorname{tr}(K)\beta_d}{\lambda} + \frac{C(\rho^2\gamma(\lambda)\log d)^2}{n}\mathbb{E}\left[\|Q\delta^{\downarrow}\|_2^2\right]$$

for a universal constant C. Note that inequality (72) provides a sufficiently tight bound on the term $\mathbb{E}\left[\|Q\delta^{\downarrow}\|_2^2\right]$. Combined with inequality (80), this completes the proof of Lemma 9.

D.1 Proof of Lemma 17

By using Jensen's inequality and then applying Cauchy-Schwarz, we find

$$\left\|\mathbb{E}\left[Q^{-1}\left(\frac{1}{n}\Phi^T\Phi - I\right)\delta^{\downarrow}\right]\right\|_2^2 \leq \left(\mathbb{E}\left[\left\|Q^{-1}\left(\frac{1}{n}\Phi^T\Phi - I\right)\delta^{\downarrow}\right\|_2\right]\right)^2$$
$$\leq \mathbb{E}\left[\left|\left|\left|Q^{-1}\left(\frac{1}{n}\Phi^T\Phi - I\right)Q^{-1}\right|\right|\right|^2\right]\mathbb{E}\left[\|Q\delta^{\downarrow}\|_2^2\right].$$

The first component of the final product can be controlled by the matrix moment bound established in the proof of inequality (48). In particular, applying (55) with $k = 2$ yields a universal constant C such that

$$\mathbb{E}\left[\left|\left|\left|Q^{-1}\left(\frac{1}{n}\Phi^T\Phi - I\right)Q^{-1}\right|\right|\right|^2\right] \leq \frac{C(\rho^2\gamma(\lambda)\log d)^2}{n},$$

which establishes the claim (83).

References

F. Bach. Sharp analysis of low-rank kernel matrix approximations. In *Proceedings of the Twenty Sixth Annual Conference on Computational Learning Theory*, 2013.

P. Bartlett, O. Bousquet, and S. Mendelson. Local Rademacher complexities. *Annals of Statistics*, 33(4):1497–1537, 2005.

P. L. Bartlett, O. Bousquet, and S. Mendelson. Localized rademacher complexities. In *Computational Learning Theory*, pages 44–58. Springer, 2002.

A. Berlinet and C. Thomas-Agnan. *Reproducing Kernel Hilbert Spaces in Probability and Statistics*. Kluwer Academic, 2004.

T. Bertin-Mahieux, D. P. Ellis, B. Whitman, and P. Lamere. The million song dataset. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR)*, 2011.

M. Birman and M. Solomjak. Piecewise-polynomial approximations of functions of the classes W_p^{α}. *Sbornik: Mathematics*, 2(3):295–317, 1967.

G. Blanchard and N. Krämer. Optimal learning rates for kernel conjugate gradient regression. In *Advances in Neural Information Processing Systems 24*, 2010.

A. Caponnetto and E. De Vito. Optimal rates for the regularized least-squares algorithm. *Foundations of Computational Mathematics*, 7(3):331–368, 2007.

R. Chen, A. Gittens, and J. A. Tropp. The masked sample covariance estimator: an analysis using matrix concentration inequalities. *Information and Inference*, to appear, 2012.

S. Fine and K. Scheinberg. Efficient SVM training using low-rank kernel representations. *Journal of Machine Learning Research*, 2:243–264, 2002.

C. Gu. *Smoothing Spline ANOVA Models*. Springer, 2002.

L. Gyorfi, M. Kohler, A. Krzyzak, and H. Walk. *A Distribution-Free Theory of Nonparametric Regression*. Springer Series in Statistics. Springer, 2002.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, 2001.

A. E. Hoerl and R. W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12:55–67, 1970.

D. Hsu, S. Kakade, and T. Zhang. Random design analysis of ridge regression. In *Proceedings of the 25nd Annual Conference on Learning Theory*, 2012.

A. Kleiner, A. Talwalkar, P. Sarkar, and M. Jordan. Bootstrapping big data. In *Proceedings of the 29th International Conference on Machine Learning*, 2012.

V. Koltchinskii. Local Rademacher complexities and oracle inequalities in risk minimization. *Annals of Statistics*, 34(6):2593–2656, 2006.

M. Ledoux and M. Talagrand. *Probability in Banach Spaces*. Springer, 1991.

D. Luenberger. *Optimization by Vector Space Methods*. Wiley, 1969.

R. McDonald, K. Hall, and G. Mann. Distributed training strategies for the structured perceptron. In *North American Chapter of the Association for Computational Linguistics (NAACL)*, 2010.

S. Mendelson. Geometric parameters of kernel machines. In *Proceedings of the Fifteenth Annual Conference on Computational Learning Theory*, pages 29–43, 2002a.

S. Mendelson. Improving the sample complexity using global data. *Information Theory, IEEE Transactions on*, 48(7):1977–1991, 2002b.

A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems 20*, 2007.

G. Raskutti, M. Wainwright, and B. Yu. Early stopping for non-parametric regression: An optimal data-dependent stopping rule. In *49th Annual Allerton Conference on Communication, Control, and Computing*, pages 1318–1325, 2011.

G. Raskutti, M. J. Wainwright, and B. Yu. Minimax-optimal rates for sparse additive models over kernel classes via convex programming. *Journal of Machine Learning Research*, 12: 389–427, March 2012.

C. Saunders, A. Gammerman, and V. Vovk. Ridge regression learning algorithm in dual variables. In *Proceedings of the 15th International Conference on Machine Learning*, pages 515–521. Morgan Kaufmann, 1998.

B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *IEEE Transactions on Information Theory*, 10(5):1299–1319, 1998.

J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

I. Steinwart, D. Hush, and C. Scovel. Optimal rates for regularized least squares regression. In *Proceedings of the 22nd Annual Conference on Learning Theory*, pages 79–93, 2009.

C. J. Stone. Optimal global rates of convergence for non-parametric regression. *Annals of Statistics*, 10(4):1040–1053, 1982.

A. B. Tsybakov. *Introduction to Nonparametric Estimation*. Springer, 2009.

S. van de Geer. *Empirical Processes in M-Estimation*. Cambridge University Press, 2000.

G. Wahba. *Spline Models for Observational Data*. CBMS-NSF Regional Conference Series in Applied Mathematics. SIAM, Philadelphia, PN, 1990.

L. Wasserman. *All of Nonparametric Statistics*. Springer, 2006.

C. Williams and M. Seeger. Using the Nyström method to speed up kernel machines. *Advances in Neural Information Processing Systems 14*, pages 682–688, 2001.

Y. Yang, M. Pilanci, and M. J. Wainwright. Randomized sketches for kernels: Fast and optimal non-parametric regression. *arXiv:1501.06195 [stat.ml]*, 2015.

Y. Yao, L. Rosasco, and A. Caponnetto. On early stopping in gradient descent learning. *Constructive Approximation*, 26(2):289–315, 2007.

T. Zhang. Leave-one-out bounds for kernel methods. *Neural Computation*, 15(6):1397–1437, 2003.

T. Zhang. Learning bounds for kernel regression using effective data dimensionality. *Neural Computation*, 17(9):2077–2098, 2005.

Y. Zhang, J. C. Duchi, and M. J. Wainwright. Communication-efficient algorithms for statistical optimization. *Journal of Machine Learning Research*, 14:3321–3363, 2013.

Journal of Machine Learning Research 16 (2015) 3341-3365 Submitted 6/14; Revised 4/15; Published 12/15

Learning Theory of Randomized Kaczmarz Algorithm

Junhong Lin JHLIN5@HOTMAIL.COM
Ding-Xuan Zhou MAZHOU@CITYU.EDU.HK
Department of Mathematics
City University of Hong Kong
83 Tat Chee Avenue
Kowloon, Hong Kong, China

Editor: Gabor Lugosi

Abstract

A relaxed randomized Kaczmarz algorithm is investigated in a least squares regression setting by a learning theory approach. When the sampling values are accurate and the regression function (conditional means) is linear, such an algorithm has been well studied in the community of non-uniform sampling. In this paper, we are mainly interested in the different case of either noisy random measurements or a nonlinear regression function. In this case, we show that relaxation is needed. A necessary and sufficient condition on the sequence of relaxation parameters or step sizes for the convergence of the algorithm in expectation is presented. Moreover, polynomial rates of convergence, both in expectation and in probability, are provided explicitly. As a result, the almost sure convergence of the algorithm is proved by applying the Borel-Cantelli Lemma.

Keywords: learning theory, relaxed randomized Kaczmarz algorithm, online learning, space of homogeneous linear functions, almost sure convergence

1. Introduction

The Kaczmarz method is an iterative projection algorithm. It was originally proposed for solving (overdetermined) systems of linear equations, and has been adapted to image reconstruction, signal processing and numerous other applications.

Given a matrix $A \in \mathbb{R}^{m\times d}$ and a vector $b \in \mathbb{R}^m$, the classical Kaczmarz algorithm (Kaczmarz, 1937) approximates a solution of the linear systems $Ax = b$ by an iterative scheme as

$$x_{k+1} = x_k + \frac{b_i - \langle a_i, x_k\rangle}{\|a_i\|^2} a_i, \tag{1}$$

where $i = k \mod m$, a_i^T is the i-th row of the matrix A, and $x_1 \in \mathbb{R}^d$ is an initial vector. Here $\langle\ ,\ \rangle$ is the inner product in $\mathbb{R}^d$ and $\|\cdot\|$ the induced norm.

The convergence of the Kaczmarz algorithm (2) is well understood (Kaczmarz, 1937), and its convergence rate depends on the order of rows of A. To avoid this dependence, a randomized Kaczmarz algorithm was considered in (Strohmer and Vershynin, 2009) by setting the probability of a row to be proportional to its norm. It takes the form

$$x_{k+1} = x_k + \frac{b_{p(i)} - \langle a_{p(i)}, x_k\rangle}{\|a_{p(i)}\|^2} a_{p(i)}, \tag{2}$$

where $p(i)$ takes values in $\{1,\dots,m\}$ with probability $\frac{\|a_{p(i)}\|^2}{\|A\|_F^2}$ with $\|A\|_F^2=\sum_{i=1}^m\sum_{j=1}^d a_{ij}^2$ being the Frobenius norm square of A. Exponential convergence rate was proved for the expected error $\mathbb{E}\|x_{k+1}-x\|^2$ of the randomized Kaczmarz algorithm (2) in (Strohmer and Vershynin, 2009). When noise exists in the sample value $b=Ax+\xi$ with ξ being a noise vector, a bound for the expected error was obtained in (Needell, 2010) and divergence was proved when the variance of ξ is positive. The error bound consists of an exponentially convergent part and a noise-driven term proportional to the noise level $\max_i \frac{|\xi_i|}{\|a_i\|^2}$.

The randomized Kaczmarz algorithm (2) was generalized in Chen and Powell (2012) to a setting with a sequence of independent random measurement vectors $\{\varphi_t\in\mathbb{R}^d\}_t$ as

$$x_{k+1}=x_k+\frac{y_k-\langle\varphi_k,x_k\rangle}{\|\varphi_k\|^2}\varphi_k. \tag{3}$$

When the measurements have no noise $y_k=\langle\varphi_k,x\rangle$, almost sure convergence was proved and quantitative error bounds were provided in (Chen and Powell, 2012).

When the linear system $Ax=b$ is overdetermined ($m>d$) and has no solution, the Kaczmarz algorithm (2) can be modified by introducing a relaxation parameter $\eta_k>0$ in front of $\frac{b_i-\langle a_i,x_k\rangle}{\|a_i\|^2}a_i$ and the output sequence $\{x_k\}$ converges to the least squares solution $\arg\min_{x\in\mathbb{R}^d}\|Ax-b\|^2$ when $\lim_{k\to\infty}\eta_k=0$. See, e.g., (Zouzias and Freris, 2013) and references therein.

Setting $\psi_k=\frac{1}{\|\varphi_k\|}\varphi_k\in\mathbb{S}^{d-1}$ and $\widetilde{y}_k=\frac{1}{\|\varphi_k\|}y_k$ yields an equivalent form of the scheme (3) as

$$x_{k+1}=x_k+\{\widetilde{y}_k-\langle\psi_k,x_k\rangle\}\,\psi_k.$$

This form is similar to those in the literature of online learning for least squares regression and together with the relaxed Kaczmarz method (Zouzias and Freris, 2013) motivates us to consider the following relaxed randomized Kaczmarz algorithm.

Definition 1 *With normalized measurement vectors $\{\psi_t\in\mathbb{S}^{d-1}\}_t$ and sample values $\{\widetilde{y}_t\in\mathbb{R}\}_t$, the relaxed randomized Kaczmarz algorithm is defined by*

$$x_{t+1}=x_t+\eta_t\{\widetilde{y}_t-\langle\psi_t,x_t\rangle\}\,\psi_t,\qquad t=1,\dots, \tag{4}$$

where $x_1\in\mathbb{R}^d$ is an initial vector and $\{\eta_t\}$ is a sequence of relaxation parameters or step sizes.

The purpose of this paper is to provide learning theory analysis for the relaxed randomized Kaczmarz algorithm. We shall assume throughout the paper that $0<\eta_t\le 2$ for each $t\in\mathbb{N}$ and that the sequence $\{z_t:=(\psi_t,\widetilde{y}_t)\}_{t\in\mathbb{N}}$ is independently drawn according to a Borel probability measure ρ on $Z:=\mathbb{S}^{d-1}\times\mathbb{R}$ which satisfies $\mathbb{E}[|\widetilde{y}|^2]<\infty$.

Our first goal is to deal with the noisy setting for the randomized Kaczmarz algorithm. When the sampling process is noisy or nonlinear (to be defined below), we show that $\{x_t\}_t$ converges to some $x^*\in\mathbb{R}^d$ in expectation if and only if $\lim_{t\to\infty}\eta_t=0$ and $\sum_{t=1}^\infty\eta_t=\infty$. Moreover, the rate of convergence in expectation cannot be too fast. It tells us that the relaxation parameter is necessary for the convergence in the noisy setting. When $\{\eta_t\}_t$ takes the form $\eta_t=\eta_1 t^{-\theta}$, we provide convergence rates in expectation and in confidence and

prove the almost sure convergence. Such results were presented in the case of no noise in (Strohmer and Vershynin, 2009; Chen and Powell, 2012) and are new in the noisy setting.

Our second goal is to give the first almost sure convergence result in online learning for least squares regression when regularization is not needed. Such a result can be found in (Tarrés and Yao, 2014) when regularization is imposed, while the convergence in expectation without regularization was proved in (Ying and Pontil, 2008). We also present the first consistency result for online learning when the approximation error (to be defined below) does not tend to zero.

2. Main Results

To introduce our learning theory approach to the relaxed randomized Kaczmarz algorithm (4), we decompose the probability measure ρ on $Z = \mathbb{S}^{d-1} \times \mathbb{R}$ into its marginal distribution ρ_X on $X := \mathbb{S}^{d-1}$ and conditional distributions $\rho(\cdot|\psi)$ at $\psi \in X$. The conditional means define the *regression function* $f_\rho : X \to \mathbb{R}$ as

$$f_\rho(\psi) = \int_{\mathbb{R}} \widetilde{y} d\rho(\widetilde{y}|\psi), \qquad \psi \in X. \tag{5}$$

The hypothesis space for the Kaczmarz algorithm (4) consists of homogeneous linear functions

$$\mathcal{H} = \left\{ f_x \in L^2_{\rho_X} : \ x \in \mathbb{R}^d \right\}, \qquad \text{where } f_x(\psi) := \langle x, \psi \rangle, \quad \psi \in X. \tag{6}$$

Definition 2 *The sampling process associated with ρ is said to be noise-free if $\widetilde{y} = f_\rho(\psi)$ almost surely. Otherwise, it is called noisy. It is said to be linear if $f_\rho \in \mathcal{H}$ as a function in $L^2_{\rho_X}$. Otherwise, it is called nonlinear.*

The main difference between our analysis in this paper and that in the literature (Strohmer and Vershynin, 2009; Needell, 2010; Chen and Powell, 2012) lies in the setting when the sampling process is either noisy or nonlinear. These two situations can be handled simultaneously by means of the least squares generalization error $\mathcal{E}(f) = \int_Z (\widetilde{y} - f(\psi))^2 d\rho$, a well developed concept in learning theory. The assumption $\mathbb{E}[|\widetilde{y}|^2] < \infty$ on ρ ensures $f_\rho \in L^2_{\rho_X}$ and $\mathcal{E}(f_\rho) < \infty$. The noise-free condition can be stated as $\mathcal{E}(f_\rho) = 0$.

It is well known that the regression function minimizes $\mathcal{E}(f)$ among all the square integral (with respect to ρ_X) functions $f \in L^2_{\rho_X}$, and satisfies

$$\mathcal{E}(f) - \mathcal{E}(f_\rho) = \|f - f_\rho\|^2_{L^2_{\rho_X}} = \int_X (f(\psi) - f_\rho(\psi))^2 d\rho_X. \tag{7}$$

Since the hypothesis space $\mathcal{H}$ is a finite dimensional subspace of $L^2_{\rho_X}$, the continuous functional $\mathcal{E}(f)$ achieves a minimizer

$$f_{\mathcal{H}} = \arg\min_{f \in \mathcal{H}} \mathcal{E}(f). \tag{8}$$

From (7) we see that $f_{\mathcal{H}}$ is the best approximation of f_ρ in the subspace $\mathcal{H}$. It is unique as the orthogonal projection of f_ρ onto $\mathcal{H}$. It can be written as $f_{\mathcal{H}} = f_{x^*}$ for some $x^* \in \mathbb{R}^d$. But such a vector x^* is not necessarily unique.

The linear condition can be stated as $f_\rho = f_{\mathcal{H}}$ or $f_\rho \in \mathcal{H}$ as functions in $L^2_{\rho_X}$. So we see that the sampling process is noisy or nonlinear if and only if $\mathcal{E}(f_{\mathcal{H}}) > 0$. Now we can state our first main result, to be proved in Section 4, which gives a characterization of the convergence of $\{x_t\}_t$ to some $x^* \in \mathbb{R}^d$ in expectation.

Theorem 3 *Define the sequence $\{x_t\}_t$ by (4). Assume $\mathcal{E}(f_{\mathcal{H}}) > 0$. Then we have the limit $\lim_{T\to\infty} \mathbb{E}_{z_1,\dots,z_T}\|x_{T+1} - x^*\|^2 = 0$ for some $x^* \in \mathbb{R}^d$ if and only if*

$$\lim_{t\to\infty} \eta_t = 0 \quad \textit{and} \quad \sum_{t=1}^{\infty} \eta_t = \infty. \tag{9}$$

In this case, we have

$$\sum_{T=1}^{\infty} \sqrt{\mathbb{E}_{z_1,\dots,z_T}\|x_{T+1} - x^*\|^2} = \infty. \tag{10}$$

Compared with the result on exponential convergence in expectation in the linear case without noise (Strohmer and Vershynin, 2009), the somewhat negative result (10) tells us that in the noisy setting the convergence in expectation cannot be as fast as $\mathbb{E}_{z_1,\dots,z_T}\|x_{T+1} - x^*\|^2 \neq O(T^{-\theta})$ for any $\theta > 2$. But for $\theta < 1$, such learning rates can be achieved by taking $\eta_t = \eta_1 t^{-\theta}$, as shown in the following second main result, to be proved in Section 4.

Theorem 4 *Let $\eta_t = \eta_1 t^{-\theta}$ for some $\theta \in (0, 1]$ and $\eta_1 \in (0, 1)$. Define the sequence $\{x_t\}_t$ by (4). Then for some $x^* \in \mathbb{R}^d$ we have*

$$\mathbb{E}_{z_1,\dots,z_T}\|x_{T+1} - x^*\|^2 \leq \begin{cases} \widetilde{C}_0 T^{-\theta}, & \textit{if } \theta < 1, \\ \widetilde{C}_0 T^{-\lambda_r \eta_1}, & \textit{if } \theta = 1, \end{cases} \tag{11}$$

where $\widetilde{C}_0$ is a constant independent of $T \in \mathbb{N}$ (given explicitly in the proof) and λ_r is the smallest positive eigenvalue of the covariance matrix C_{ρ_X} of the probability measure ρ_X defined by

$$C_{\rho_X} = \mathbb{E}_{\rho_X}[\psi\psi^T] = \int_X \psi\psi^T d\rho_X. \tag{12}$$

Our third main result is the following confidence-based estimate for the error which will be proved in Section 5.

Theorem 5 *Assume that for some constant $M > 0$, $|\widetilde{y}| \leq M$ almost surely. Let $\theta \in [1/2, 1]$, $\eta_t = \eta_1 t^{-\theta}$ with $0 < \eta_1 < \min\{1, \frac{1}{2\lambda_r}\}$, and $2 \leq T \in \mathbb{N}$. Then for some $x^* \in \mathbb{R}^d$ and for any $0 < \delta < 1$, with confidence at least $1 - \delta$ we have*

$$\|x_{T+1} - x^*\| \leq \begin{cases} \widetilde{C}_1 T^{-\theta/2} \left(\log \frac{4}{\delta}\right)^2 \log T, & \textit{when } \theta \in [1/2, 1), \\ \widetilde{C}_1 T^{-\lambda_r \eta_1} \log \frac{2}{\delta} \sqrt{\log T}, & \textit{when } \theta = 1, \end{cases} \tag{13}$$

where $\widetilde{C}_1$ is a positive constant independent of T or δ (given explicitly in the proof).

Our last main result is about the almost sure convergence of the algorithm, which will be proved in Section 6.

Theorem 6 *Under the assumptions of Theorem 5, we have for any $\epsilon \in (0,1]$, the following holds for some $x^* \in \mathbb{R}^d$:*

(A) When $1/2 \le \theta < 1$, $\lim_{t\to\infty} t^{\theta(1-\epsilon)/2}\|x_{t+1} - x^*\| = 0$ almost surely.

(B) When $\theta = 1$, $\lim_{t\to\infty} t^{\lambda_r \eta_1 (1-\epsilon)}\|x_{t+1} - x^*\| = 0$ almost surely.

Let us demonstrate our setting by two examples without noise considered in the literature. The first example appeared in (Chen and Powell, 2012).

Example 1 *If random measurement vectors $\{\varphi_t\}_{t=1}^{\infty}$ are independent and nonzero almost surely, then $\{\psi_k = \frac{1}{\|\varphi_k\|}\varphi_k \in \mathbb{S}^{d-1}\}$ are independent.*

The second example is from (Strohmer and Vershynin, 2009).

Example 2 *Define the random vector φ which is a normalized row of a full rank matrix $A \in \mathbb{R}^{m\times d}$, with probabilities as*

$$\varphi = \frac{a_j}{\|a_j\|} \quad \text{with probability} \quad \frac{\|a_j\|^2}{\|A\|_F^2} \quad j = 1, \cdots, m.$$

It was shown in Strohmer and Vershynin (2009) that the smallest eigenvalue of the covariance matrix is positive:

$$\lambda_{\min}(\mathbb{E}[\varphi\varphi^T]) \ge \frac{1}{\|A\|_F^2 \|A^{-1}\|^2}.$$

It means $r = d$ and $\lambda_r \ge \frac{1}{\|A\|_F^2\|A^{-1}\|^2}$.

The third example is on homoskedastic models (Johnston, 1963).

Example 3 *In the literature of homoskedastic models, it is assumed that the sample value $\{y_t\}_t$ satisfies $y_t = \langle x^*, \psi_t\rangle + \xi_t$ with $\{\xi_t\}_t$ being independently drawn according to a zero mean probability measure ξ. This corresponds to the special case when the conditional distributions $\rho(\cdot|\psi)$ are given by $\rho(\cdot|\psi) = f_\rho(\psi) + \xi$. Our setting induced by ρ is more general and allows heteroskedastic models.*

3. Connections to Learning Theory

The relaxed randomized Kaczmarz algorithm defined by (4) may be rewritten as an online learning algorithm with output functions from the hypothesis space (6), and our main results stated in the last section are new even in the online learning literature. To demonstrate this, we denote the tth output function F_t on X induced by the vector x_t to be given by $F_t(\psi) = \langle x_t, \psi\rangle$ for $\psi \in X$. Then the iteration relation (4) gives

$$F_{t+1} = F_t + \eta_t \{\widetilde{y}_t - F_t(\psi_t)\} \langle \cdot, \psi_t\rangle. \tag{14}$$

This is a special kernel-based least squares online learning algorithm. Here a (Mercer) kernel on a metric space $\mathcal{X}$ means a function $K : \mathcal{X}\times\mathcal{X} \to \mathbb{R}$ which is continuous, symmetric and the matrix $(K(x_i, x_j))_{i,j=1}^{\ell}$ is positive semidefinite for any finite subset $\{x_i\}_{i=1}^{\ell} \subseteq \mathcal{X}$.

It generates a reproducing kernel Hilbert space $(\mathcal{H}_K, \|\cdot\|_K)$ by the set of fundamental functions $\{K(\cdot,x) : x \in \mathcal{X}\}$ with the inner product $\langle K(\cdot,x), K(\cdot,y)\rangle_K = K(x,y)$. A least squares regularized online learning algorithm in $\mathcal{H}_K$ is defined with $\{(\psi_t, \widetilde{y}_t) \in \mathcal{X} \times \mathbb{R}\}_t$ drawn independently according to a probability measure on $Z = \mathcal{X} \times \mathbb{R}$ as

$$F_{t+1} = F_t - \eta_t \left\{ (F_t(\psi_t) - \widetilde{y}_t) K(\cdot, \psi_t) + \lambda F_t \right\}, \qquad t = 1, \ldots, \tag{15}$$

where $\lambda \geq 0$ is a regularization parameter. The consistency of the online learning algorithm (15) is well understood when the approximation error $\mathcal{D}(\lambda)$ tends to zero as $\lambda \to 0$.

Definition 7 *The approximation error (or regularization error) of the pair (ρ, K) is defined for $\lambda > 0$ as*

$$\mathcal{D}(\lambda) = \inf_{f \in \mathcal{H}_K} \left\{ \mathcal{E}(f) - \mathcal{E}(f_\rho) + \lambda \|f\|_K^2 \right\} = \inf_{f \in \mathcal{H}_K} \left\{ \|f - f_\rho\|_{L^2_{\rho_\mathcal{X}}}^2 + \lambda \|f\|_K^2 \right\}. \tag{16}$$

When $\lambda > 0$ (with regularization) and $\lim_{\lambda \to 0} \mathcal{D}(\lambda) = 0$, the error $\|F_{T+1} - f_\rho\|_{L^2_{\rho_\mathcal{X}}}^2$ in expectation and in confidence was bounded in (Smale and Yao, 2005; Ying and Zhou, 2006; Smale and Zhou, 2009; Tarrés and Yao, 2014) by means of the decay of $\mathcal{D}(\lambda)$ and T. The error analysis was done in Ying and Pontil (2008) without regularization ($\lambda = 0$) but under the approximation error condition $\lim_{\lambda \to 0} \mathcal{D}(\lambda) = 0$. The error $\|F_{T+1} - f_\rho\|_K^2$ with the $\mathcal{H}_K$-metric was also analyzed when $f_\rho \in \mathcal{H}_K$.

If we take the kernel to be the linear one: $K(x,y) = \langle x, y\rangle$ with $\mathcal{X} = \mathbb{R}^d$, and assume that the marginal distribution $\rho_\mathcal{X}$ is supported on $X = \mathbb{S}^{d-1}$, then $\psi_t \in \mathbb{S}^{d-1}$ almost surely. Set $\lambda = 0$, we see that the relaxed randomized Kaczmarz algorithm expressed in the form (14) is the least squares online learning algorithm (15) without regularization. So the error analysis from Ying and Pontil (2008) applies, but the condition $\lim_{\lambda \to 0} \mathcal{D}(\lambda) = 0$ is required for the consistency in $L^2_{\rho_\mathcal{X}}$ and even stronger conditions (stronger than $f_\rho \in \mathcal{H}_K$) are needed for the consistency in the $\mathcal{H}_K$-metric.

Notice that for the linear kernel, $\|x\| = \|\langle \cdot, x\rangle\|_K$. So the error analysis carried out in this paper provides bounds for the error $\|F_{T+1} - f_\mathcal{H}\|_K$ without the condition $\lim_{\lambda \to 0} \mathcal{D}(\lambda) = 0$. Such results cannot be found in the literature of online learning. It leads to the problem of carrying our similar error analysis for more general online learning algorithms associated with more general kernels. Moreover, the best convergence rate in expectation of the general kernel-based least squares online learning algorithm is $O(T^{-1/2})$ in the literature (Smale and Yao, 2005; Ying and Zhou, 2006; Smale and Zhou, 2009; Tarrés and Yao, 2014; Hu et al., 2015). Theorem 4 demonstrates that the special online learning algorithm (4) has convergence rates of type $O(T^{-(1-\epsilon)})$ for any $\epsilon > 0$ and even of type $O(T^{-1} \log T)$ shown in Theorem 8 below, which is a great improvement.

Note that there is a gap between the negative result (10) and the positive one (11), which leads to the natural question whether learning rates of type $\mathbb{E}_{z_1,\ldots,z_T} \|x_{T+1} - x^*\|^2 = O(T^{-\theta})$ are possible for $1 < \theta \leq 2$. We conjecture that this is impossible for a general probability measure ρ, but a noise condition might help. The case $\theta = 1$ with a slight logarithmic modification $O(T^{-1} \log T)$ can be achieved by imposing a minor restriction on the step size in the following theorem which will be proved in the next section. The authors thank Dr. Yiming Ying for pointing out this result.

Theorem 8 *Let λ_r be as in Theorem 4 and $\eta_t = \frac{1}{\lambda_r(t+t_0)}$ for some $t_0 \in \mathbb{N}$ such that $t_0\lambda_r \geq 1$. Define the sequence $\{x_t\}_t$ by (4). Then for some $x^* \in \mathbb{R}^d$, we have*

$$\mathbb{E}_{z_1,\cdots,z_T}\|x_{T+1} - x^*\|^2 \leq \widetilde{C}_3(T+t_0)^{-1}\log T,$$

where $\widetilde{C}_3$ is a constant independent of $T \in \mathbb{N}$ (given explicitly in the proof).

4. Convergence in Expectation

In this section we prove our main results on convergence in expectation. To this end, we need some preliminary analysis.

Recall the function $f_{\mathcal{H}}$ defined by (8). It equals f_{x^*} for some $x^* \in \mathbb{R}^d$. As the orthogonal projection of f_ρ onto the finite dimensional subspace $\mathcal{H}$ in the Hilbert space $L^2_{\rho_X}$, it satisfies

$$\langle f_\rho - f_{x^*}, f_x\rangle_{L^2_{\rho_X}} = \int_X (f_\rho(\psi) - \langle x^*, \psi\rangle)\langle x, \psi\rangle d\rho_X(\psi) = 0, \quad \forall x \in \mathbb{R}^d. \tag{17}$$

The vector x^* is not necessarily unique. To see this, we use the covariance matrix C_{ρ_X} of the measure ρ_X defined by (12) and denote its eigenvalues to be $\lambda_1 \geq \ldots \geq \lambda_r > \lambda_{r+1} = \ldots = \lambda_d = 0$ where $r \in \{1,\ldots,d\}$ is the rank of C_{ρ_X}. Denote the eigenspace of C_{ρ_X} associated with the eigenvalue 0 as V_0 and the orthogonal projection onto V_0 as P_0. Then any vector $x^* + v$ from the set $x^* + V_0$ is also a minimizer of $\mathcal{E}(f_x)$ in $\mathbb{R}^d$, but $f_{x^*+v} = f_{x^*} = f_{\mathcal{H}}$ as functions in the space $L^2_{\rho_X}$.

The following lemma about the residual vectors $\{r_t = x_t - x^*\}_t$ is a crucial step in our analysis in this section.

Lemma 9 *Define the sequence $\{x_t\}_t$ by (4). Let $x^* \in \mathbb{R}^d$ be such that $f_{x^*} = f_{\mathcal{H}}$. Denote $r_t = x_t - x^*$. Then there holds*

$$\mathbb{E}_{z_t}[\|r_{t+1}\|^2] = \|r_t\|^2 + (-2\eta_t + \eta_t^2)\|f_{r_t}\|^2_{L^2_{\rho_X}} + \eta_t^2\mathcal{E}(f_{\mathcal{H}}), \qquad \forall\ t \in \mathbb{N}. \tag{18}$$

Proof Subtract x^* from both sides of (4) and take inner products. We see from $\|\psi_t\| = 1$ that

$$\|r_{t+1}\|^2 = \|r_t\|^2 + 2\eta_t\left\{\widetilde{y}_t - \langle\psi_t, x_t\rangle\right\}\langle\psi_t, r_t\rangle + \eta_t^2\left\{\widetilde{y}_t - \langle\psi_t, x_t\rangle\right\}^2. \tag{19}$$

Since x_t does not depend on z_t, taking expectation with respect to z_t, we see from $\mathbb{E}[\widetilde{y}_t|\psi_t] = f_\rho(\psi_t)$ and $\mathbb{E}_{z_t}\left\{\widetilde{y}_t - \langle\psi_t, x_t\rangle\right\}^2 = \mathcal{E}(f_{x_t})$ that

$$\mathbb{E}_{z_t}[\|r_{t+1}\|^2] = \|r_t\|^2 + 2\eta_t\mathbb{E}_{\psi_t}\left[\left\{f_\rho(\psi_t) - \langle\psi_t, x_t\rangle\right\}\langle\psi_t, r_t\rangle\right] + \eta_t^2\mathcal{E}(f_{x_t}).$$

By (17), we know that the middle term above equals

$$2\eta_t\mathbb{E}_{\psi_t}\left[\left\{\langle\psi_t, x^*\rangle - \langle\psi_t, x_t\rangle\right\}\langle\psi_t, r_t\rangle\right] = 2\eta_t\mathbb{E}_{\psi_t}\left[\left\{\langle\psi_t, -r_t\rangle\right\}\langle\psi_t, r_t\rangle\right] = -2\eta_t\|f_{r_t}\|^2_{L^2_{\rho_X}}.$$

Since f_{x^*} is the orthogonal projection of f_ρ onto $\mathcal{H}$, there holds $\mathcal{E}(f_{x_t}) = \mathcal{E}(f_\rho) + \|f_\rho - f_{x^*}\|^2_{L^2_{\rho_X}} + \|f_{x^*} - f_{x_t}\|^2_{L^2_{\rho_X}} = \mathcal{E}(f_{\mathcal{H}}) + \|f_{r_t}\|^2_{L^2_{\rho_X}}$. Then the desired identity (18) follows. ■

We are in a position to prove our first main result.

Proof of Theorem 3 *Necessity.* We first analyze the first two terms of the right hand side of the identity (18) in Lemma 9. Since $0 < \eta_t \leq 2$, we have $-2\eta_t + \eta_t^2 < 0$. Observe from the Schwarz inequality that $|f_{r_t}(\psi)|^2 = |\langle r_t, \psi\rangle|^2 \leq \|r_t\|^2\|\psi\|^2 = \|r_t\|^2$ and thereby $\|f_{r_t}\|^2_{L^2_{\rho_X}} \leq \|r_t\|^2$. It follows that

$$\|r_t\|^2 + (-2\eta_t + \eta_t^2)\|f_{r_t}\|^2_{L^2_{\rho_X}} \geq \|r_t\|^2 + (-2\eta_t + \eta_t^2)\|r_t\|^2 = (1-\eta_t)^2\|r_t\|^2.$$

This together with (18) implies

$$\mathbb{E}_{z_t}[\|r_{t+1}\|^2] \geq (1-\eta_t)^2\|r_t\|^2 + \eta_t^2\mathcal{E}(f_{\mathcal{H}}). \tag{20}$$

Then we can proceed with proving the necessity. If $\lim_{T\to\infty}\mathbb{E}_{z_1,\ldots,z_T}\|x_{T+1} - x^*\|^2 = 0$ for some $x^* \in \mathbb{R}^d$ and $\mathcal{E}(f_{\mathcal{H}}) > 0$, we know from (20) that $\lim_{T\to\infty}\eta_T = 0$. It ensures the existence of some integer $t_0 \geq 2$ such that $\eta_t \leq \frac{1}{3}$ for any $t \geq t_0$. Since $1 - \eta \geq \exp\{-2\eta\}$ for $0 < \eta \leq \frac{1}{3}$, we know that for any $t \geq t_0$, $(1-\eta_t)^2 \geq \exp\{-4\eta_t\}$. Combining this with (20) yields

$$\mathbb{E}_{z_1,\ldots,z_T}\|x_{T+1} - x^*\|^2 \geq \Pi_{t=t_0}^T \exp\{-4\eta_t\}\,\mathbb{E}_{z_1,\ldots,z_{t_0-1}}\|r_{t_0}\|^2.$$

But (20) also tells us that $\mathbb{E}_{z_1,\ldots,z_{t_0-1}}\|r_{t_0}\|^2 \geq \eta_{t_0-1}^2\mathcal{E}(f_{\mathcal{H}}) > 0$. So

$$\mathbb{E}_{z_1,\ldots,z_T}\|x_{T+1} - x^*\|^2 \geq \exp\left\{-4\sum_{t=t_0}^T \eta_t\right\}\eta_{t_0-1}^2\mathcal{E}(f_{\mathcal{H}}).$$

Since $\lim_{T\to\infty}\mathbb{E}_{z_1,\ldots,z_T}\|x_{T+1} - x^*\|^2 = 0$, we must have $\sum_{t=1}^\infty \eta_t = \infty$. This proves the necessity.

Sufficiency. Recall that V_0 is the eigenspace of the covariance matrix C_{ρ_X} associated with the eigenvalue 0 and P_0 is the orthogonal projection onto V_0. Then ψ_t is orthogonal to V_0 almost surely for each t. It follows that $P_0(x_{t+1}) = P_0(x_t)$ and thereby $P_0(x_t) = P_0(x_1)$ for each t. Take the vector x^* to be the minimizer of $\mathcal{E}(f_x)$ in $\mathbb{R}^d$ such that $P_0(x^*) = P_0(x_1)$. With this choice, r_t is orthogonal to V_0 for each t, and belongs to the orthogonal complement $V_0^\perp$. Note that the eigenvalues of C_{ρ_X} restricted to the subspace $V_0^\perp$ is at least $\lambda_r > 0$. So we have

$$\|f_{r_t}\|^2_{L^2_{\rho_X}} = \int_X |\langle\psi, r_t\rangle|^2\,d\rho_X = \int_X r_t^T\psi\psi^T r_t d\rho_X = r_t^T C_{\rho_X} r_t \tag{21}$$

and $\|f_{r_t}\|^2_{L^2_{\rho_X}} \geq \lambda_r\|r_t\|^2$. The condition $\lim_{t\to\infty}\eta_t = 0$ ensures the existence of some $t_1 \in \mathbb{N}$ such that $\eta_t \leq 1$ for any $t \geq t_1$. Thus, we see from (18) in Lemma 9 that for $t \geq t_1$,

$$\mathbb{E}_{z_t}[\|r_{t+1}\|^2] \leq \|r_t\|^2 - \eta_t\|f_{r_t}\|^2_{L^2_{\rho_X}} + \eta_t^2\mathcal{E}(f_{\mathcal{H}}) \leq (1 - \eta_t\lambda_r)\|r_t\|^2 + \eta_t^2\mathcal{E}(f_{\mathcal{H}}).$$

Applying this inequality iteratively for $t = T, \cdots t_1$ yields

$$\mathbb{E}_{z_1,\ldots,z_T}[\|r_{T+1}\|^2] \leq \mathbb{E}_{z_1,\ldots,z_{t_1-1}}[\|r_{t_1}\|^2]\prod_{t=t_1}^T(1-\eta_t\lambda_r) + \mathcal{E}(f_{\mathcal{H}})\sum_{t=t_1}^T\eta_t^2\prod_{k=t+1}^T(1-\eta_k\lambda_r), \tag{22}$$

where we denote $\prod_{k=t+1}^{T}(1-\eta_k\lambda_r)=1$ for $t=T$. By the condition $\sum_{t=1}^{\infty}\eta_t=\infty$, one has

$$\prod_{t=t_1}^{T}(1-\eta_t\lambda_r)\le\exp\left\{-\lambda_r\sum_{t=t_1}^{T}\eta_t\right\}\to 0\quad\text{as } T\to\infty.$$

Thus for any $\varepsilon>0$, there exists $t_2=t_2(\varepsilon)\in\mathbb{N}$ such that for any $T\ge t_2$,

$$\mathbb{E}_{z_1,\ldots,z_{t_1-1}}[\|r_{t_1}\|^2]\prod_{t=t_1}^{T}(1-\eta_t\lambda_r)\le\varepsilon.$$

To deal with the other term of the bound (22) for $\|r_{T+1}\|^2$, we use the assumption $\lim_{t\to\infty}\eta_t=0$, and find some integer $t(\varepsilon)\ge t_1$ such that $\eta_t\le\lambda_r\varepsilon$ for any $t\ge t(\varepsilon)$. Write

$$\sum_{t=t_1}^{T}\eta_t^2\prod_{k=t+1}^{T}(1-\eta_k\lambda_r)=\sum_{t=t_1}^{t(\varepsilon)}\eta_t^2\prod_{k=t+1}^{T}(1-\eta_k\lambda_r)+\sum_{t=t(\varepsilon)+1}^{T}\eta_t^2\prod_{k=t+1}^{T}(1-\eta_k\lambda_r).\tag{23}$$

The second term of (23) can be bounded as

$$\begin{aligned}\sum_{t=t(\varepsilon)+1}^{T}\eta_t^2\prod_{k=t+1}^{T}(1-\eta_k\lambda_r)&=\varepsilon\sum_{t=t(\varepsilon)+1}^{T}\eta_t\lambda_r\prod_{k=t+1}^{T}(1-\eta_k\lambda_r)\\&=\varepsilon\sum_{t=t(\varepsilon)+1}^{T}(1-(1-\eta_t\lambda_r))\prod_{k=t+1}^{T}(1-\eta_k\lambda_r)\\&=\varepsilon\left(1-\prod_{k=t(\varepsilon)+1}^{T}(1-\eta_k\lambda_r)\right)\le\varepsilon.\end{aligned}$$

To bound the first term of (23), we apply the condition $\sum_{t=1}^{\infty}\eta_t=\infty$ again and find some integer $t_3=t_3(\varepsilon)>t(\varepsilon)$ such that $\sum_{k=t(\varepsilon)+1}^{t_3}\eta_k\ge\frac{1}{\lambda_r}\log\frac{t(\varepsilon)}{\varepsilon}$. Hence

$$\sum_{k=t(\varepsilon)+1}^{T}\eta_k\ge\sum_{k=t(\varepsilon)+1}^{t_3}\eta_k\ge\frac{1}{\lambda_r}\log\frac{t(\varepsilon)}{\varepsilon},\qquad\forall\ T\ge t_3.$$

It thus follows that for each $t\in\{t_1,\ldots,t(\varepsilon)\}$,

$$\prod_{k=t+1}^{T}(1-\eta_k\lambda_r)\le\exp\left\{-\lambda_r\sum_{k=t+1}^{T}\eta_k\right\}\le\exp\left\{-\lambda_r\sum_{k=t(\varepsilon)+1}^{T}\eta_k\right\}\le\frac{\varepsilon}{t(\varepsilon)}.$$

Combining with the fact $\eta_t\le 1$ for each $t\ge t_1$, we see that the first term of (23) can be bounded as

$$\sum_{t=t_1}^{t(\varepsilon)}\eta_t^2\prod_{k=t+1}^{T}(1-\eta_k\lambda_r)\le\frac{\varepsilon}{t(\varepsilon)}\sum_{t=t_1}^{t(\varepsilon)}\eta_t^2\le\varepsilon.$$

From the above analysis, we know that when $T \geq \max\{t_1, t(\varepsilon), t_2, t_3\}$,

$$\mathbb{E}_{z_1,\ldots,z_T}[\|r_{T+1}\|^2] \leq \varepsilon + 2\mathcal{E}(f_{\mathcal{H}})\varepsilon.$$

This proves the convergence $\lim_{T\to\infty} \mathbb{E}_{z_1,\ldots,z_T}\|x_{T+1} - x^*\|^2 = 0$ for some $x^* \in \mathbb{R}^d$ and the sufficiency is verified.

From the bound (20), we also see that

$$\mathbb{E}_{z_1,\ldots,z_T}\|x_{T+1} - x^*\|^2 \geq \eta_T^2 \mathcal{E}(f_{\mathcal{H}}), \qquad \forall\, T \in \mathbb{N}.$$

This implies that

$$\sum_{T=1}^{\infty} \sqrt{\mathbb{E}_{z_1,\ldots,z_T}\|x_{T+1} - x^*\|^2} \geq \sqrt{\mathcal{E}(f_{\mathcal{H}})} \sum_{T=1}^{\infty} \eta_T = \infty.$$

The proof of Theorem 3 is complete. ■

In the proof of our second main result, we need some elementary inequalities.

Lemma 10 *(a) For $\nu, a > 0$, there holds*

$$\exp\{-\nu x\} \leq \left(\frac{a}{\nu \mathrm{e}}\right)^a x^{-a}, \qquad \forall x > 0. \tag{24}$$

(b) Let $\nu > 0$ and $q_2 \geq 0$. If $0 < q_1 < 1$, then for any $t \in \mathbb{N}$, we have

$$\sum_{i=1}^{t-1} i^{-q_2} \exp\left\{-\nu \sum_{j=i+1}^{t} j^{-q_1}\right\} \leq \left(\frac{2^{q_1+q_2}}{\nu} + \left(\frac{1+q_2}{\nu(1-2^{q_1-1})\mathrm{e}}\right)^{\frac{1+q_2}{1-q_1}}\right) t^{q_1-q_2}. \tag{25}$$

For $q_1 = 1$, we have

$$\sum_{i=1}^{t-1} i^{-q_2} \exp\left\{-\nu \sum_{j=i+1}^{t} j^{-1}\right\} \leq \begin{cases} \frac{2^{q_2}}{|\nu-q_2+1|} t^{-\min\{\nu, q_2-1\}}, & \text{if } \nu \neq q_2 - 1, \\ 2^{q_2} t^{-\nu} \log t, & \text{if } \nu = q_2 - 1. \end{cases} \tag{26}$$

(c) For any $t < T \in \mathbb{N}$ and $\theta \in (0, 1]$, there holds

$$\sum_{k=t+1}^{T} k^{-\theta} \geq \begin{cases} \frac{1}{1-\theta}[(T+1)^{1-\theta} - (t+1)^{1-\theta}], & \text{if } \theta < 1, \\ \log(T+1) - \log(t+1), & \text{if } \theta = 1. \end{cases} \tag{27}$$

(d) For $\theta \in (0, 1]$, $\mu > 0$, and $T \in \mathbb{N}$, there holds

$$\exp\left\{-\mu \sum_{t=1}^{T} t^{-\theta}\right\} \leq \begin{cases} \exp\left\{\frac{\mu}{1-\theta}\right\} \left(\frac{\theta}{\mu \mathrm{e}}\right)^{\frac{\theta}{1-\theta}} T^{-\theta}, & \text{if } \theta < 1, \\ T^{-\mu}, & \text{if } \theta = 1. \end{cases} \tag{28}$$

Proof The inequalities in parts (a) and (b) can be found in (Smale and Zhou, 2009, Lemma 2).

Part (c) can be proved by noting that

$$\sum_{k=t+1}^{T} k^{-\theta} \geq \sum_{k=t+1}^{T} \int_{k}^{k+1} x^{-\theta} dx = \int_{t+1}^{T+1} x^{-\theta} dx.$$

For part (d), we use the inequality (27) in part (c) to derive

$$\exp\left\{-\mu \sum_{t=1}^{T} t^{-\theta}\right\} \leq \begin{cases} \exp\left\{\frac{\mu}{1-\theta}\right\} \exp\left\{-\frac{\mu}{1-\theta} T^{1-\theta}\right\}, & \text{if } \theta < 1, \\ T^{-\mu}, & \text{if } \theta = 1. \end{cases}$$

For $\theta \in (0,1)$, by applying (24) with $\nu = \frac{\mu}{1-\theta}$, $x = T^{1-\theta}$ and $a = \frac{\theta}{1-\theta}$, we get

$$\exp\left\{-\frac{\mu}{1-\theta} T^{1-\theta}\right\} \leq \left(\frac{\theta}{\mu \mathrm{e}}\right)^{\frac{\theta}{1-\theta}} T^{-\theta}.$$

This proves the result. ■

We can now prove our second main result. This is done by following the estimate (22) in the proof of Theorem 3.

Proof of Theorem 4 Since $\eta_1 < 1$, we have $\eta_t < 1$ for all $t \in \mathbb{N}$. Therefore, we can take $t_1 = 1$ in (22) and obtain

$$\begin{aligned} \mathbb{E}_{z_1,\ldots,z_T}[\|r_{T+1}\|^2] &\leq \|r_1\|^2 \prod_{t=1}^{T} (1 - \eta_t \lambda_r) + \mathcal{E}(f_{\mathcal{H}}) \sum_{t=1}^{T} \eta_t^2 \prod_{k=t+1}^{T} (1 - \eta_k \lambda_r) \\ &\leq \|r_1\|^2 \exp\left\{-\lambda_r \eta_1 \sum_{t=1}^{T} t^{-\theta}\right\} \\ &\quad + \mathcal{E}(f_{\mathcal{H}}) \eta_1^2 \sum_{t=1}^{T} t^{-2\theta} \exp\left\{-\lambda_r \eta_1 \sum_{k=t+1}^{T} k^{-\theta}\right\}. \end{aligned} \tag{29}$$

Applying part (d) with $\mu = \lambda_r \eta_1$ of Lemma 10, we know that the first term of (29) can be bounded as

$$\|r_1\|^2 \exp\left\{-\lambda_r \eta_1 \sum_{t=1}^{T} t^{-\theta}\right\} \leq \begin{cases} \|r_1\|^2 \exp\left\{\frac{\lambda_r \eta_1}{1-\theta}\right\} \left(\frac{\theta}{\lambda_r \eta_1 \mathrm{e}}\right)^{\frac{\theta}{1-\theta}} T^{-\theta}, & \text{if } \theta < 1, \\ \|r_1\|^2 T^{-\lambda_r \eta_1}, & \text{if } \theta = 1. \end{cases}$$

Applying part (b) of Lemma 10 with $q_1 = \theta, q_2 = 2\theta$, $\nu = \lambda_r \eta_1$, and noting that $\lambda_r \eta_1 < 1$ by $\eta_1 \in (0,1)$ and $\lambda_r \in (0,1]$, we know that the second term of (29) can be bounded by

$$\begin{cases} \mathcal{E}(f_{\mathcal{H}}) \eta_1^2 \left(1 + \frac{2^{3\theta}}{\lambda_r \eta_1} + \left(\frac{1+2\theta}{\lambda_r \eta_1 (1-2^{\theta-1}) \mathrm{e}}\right)^{\frac{1+2\theta}{1-\theta}}\right) T^{-\theta}, & \text{if } \theta < 1, \\ \mathcal{E}(f_{\mathcal{H}}) \eta_1^2 \left(1 + \frac{4}{1-\lambda_r \eta_1}\right) T^{-\lambda_r \eta_1}, & \text{if } \theta = 1. \end{cases}$$

Thus, we get our desired result with $\widetilde{C}_0$ given by

$$\widetilde{C}_0 = \begin{cases} \|r_1\|^2 \exp\left\{\frac{\lambda_r\eta_1}{1-\theta}\right\}\left(\frac{\theta}{\lambda_r\eta_1 \mathrm{e}}\right)^{\frac{\theta}{1-\theta}} \\ \qquad +\mathcal{E}(f_{\mathcal{H}})\eta_1^2\left(\frac{\lambda_r\eta_1+2^{3\theta}}{\lambda_r\eta_1}+\left(\frac{1+2\theta}{\lambda_r\eta_1(1-2^{\theta-1})\mathrm{e}}\right)^{\frac{1+2\theta}{1-\theta}}\right), & \text{if } \theta<1, \\ \|r_1\|^2+\mathcal{E}(f_{\mathcal{H}})\eta_1^2\left(1+\frac{4}{1-\lambda_r\eta_1}\right), & \text{if } \theta=1. \end{cases}$$

This completes the proof of Theorem 4. ■

Remark 11 *From the proof of Theorem 4, we see that if $\mathcal{E}(f_{\mathcal{H}})=0$, then for some $x^*\in\mathbb{R}^d$, we have*

$$\mathbb{E}_{z_1,\ldots,z_T}[\|r_{T+1}\|^2] \le \|r_1\|^2\prod_{t=1}^{T}(1-\eta_t\lambda_r).$$

The above argument actually can be used to prove Theorem 8.

Proof of Theorem 8 Since $\eta_1\le 1$, we have $\eta_t\le 1$ for all $t\in\mathbb{N}$. Thus, we can take $t_1=1$ in (22) and obtain

$$\begin{aligned} \mathbb{E}_{z_1,\ldots,z_T}[\|r_{T+1}\|^2] &\le \|r_1\|^2\prod_{t=1}^{T}(1-\eta_t\lambda_r)+\mathcal{E}(f_{\mathcal{H}})\sum_{t=1}^{T}\eta_t^2\prod_{k=t+1}^{T}(1-\eta_k\lambda_r) \\ &= \|r_1\|^2\prod_{t=1}^{T}\left(1-\frac{1}{t+t_0}\right) \\ &\quad +\frac{\mathcal{E}(f_{\mathcal{H}})}{\lambda_r^2}\sum_{t=1}^{T}\frac{1}{(t+t_0)^2}\prod_{k=t+1}^{T}\left(1-\frac{1}{k+t_0}\right). \end{aligned}$$

We note that

$$\prod_{k=t+1}^{T}\left(1-\frac{1}{k+t_0}\right)=\prod_{k=t+1}^{T}\frac{k+t_0-1}{k+t_0}=\frac{t+t_0}{T+t_0}.$$

It thus follows that

$$\mathbb{E}_{z_1,\ldots,z_T}[\|r_{T+1}\|^2] \le \|r_1\|^2\frac{t_0}{T+t_0}+\frac{\mathcal{E}(f_{\mathcal{H}})}{\lambda_r^2}\frac{1}{T+t_0}\sum_{t=1}^{T}\frac{1}{t+t_0}.$$

With $\sum_{t=1}^{T}\frac{1}{t+t_0}\le\log\frac{T+t_0}{t_0+1}\le\log T$, we get the desired result with $\widetilde{C}_3$ given by

$$\widetilde{C}_3=t_0\|r_1\|^2+\frac{\mathcal{E}(f_{\mathcal{H}})}{\lambda_r^2}.$$

This proves Theorem 8. ■

5. Confidence-Based Estimates for Convergence

In this section, we prove our third main result, Theorem 5. Recall that V_0 is the eigenspace of the covariance matrix C_{ρ_X} associated with the eigenvalue 0. We choose x^* as in the proof of the sufficiency part of Theorem 3. With this choice, r_t belongs to the orthogonal complement $V_0^\perp$ almost surely. Our error analysis is based on the following error decomposition.

5.1 Error Decomposition

For $t \in \mathbb{N}$, set the operator $\Pi_k^t = \prod_{j=k}^t (I - \eta_j C_{\rho_X})$ on $\mathbb{R}^d$ for $k \le t$ and $\Pi_{t+1}^t = I$. Subtracting x^* from both sides of (4), we have

$$r_{k+1} = (I - \eta_k C_{\rho_X}) r_k + \eta_k \chi_k, \tag{30}$$

where

$$\chi_k = (\tilde{y}_k - \langle \psi_k, x^* \rangle)\psi_k + (C_{\rho_X} - \psi_k \psi_k^T) r_k.$$

Applying this relationship iteratively for $k = t, \cdots, 1$, we get

$$r_{t+1} = \Pi_1^t r_1 + \sum_{k=1}^t \eta_k \Pi_{k+1}^t \chi_k.$$

Thus

$$\|r_{t+1}\| \le \left\|\Pi_1^t r_1\right\| + \left\| \sum_{k=1}^t \eta_k \Pi_{k+1}^t \chi_k \right\|. \tag{31}$$

The first term of the bound (31) is caused by the initial error, which is deterministic and will be estimated in subsection 5.2. The second term is the sample error depending on the sample. Since r_k is independent of z_k, by $\mathbb{E}[\tilde{y}_k | \psi_k] = f_\rho(\psi_k)$ and (17),

$$\mathbb{E}[\chi_k | z_1, \ldots, z_{k-1}] = \int_X (f_\rho(\psi) - \langle x^*, \psi \rangle)\psi + (C_{\rho_X} - \psi\psi^T) r_k d\rho_X(\psi) = 0.$$

It tells us that $\{\omega_k := \eta_k \Pi_{k+1}^t \chi_k\}_k$ is a martingale difference sequence. The idea of analyzing the sample error by properties of martingale difference sequences can be found in the recent work in (Tarrés and Yao, 2014) to which details about martingale difference sequences are referred. In particular, we can apply the following Pinelis-Bernstein inequality from (Tarrés and Yao, 2014) (derived from (Pinelis, 1994, Theorem 3.4)) to estimate the sample error.

Lemma 12 *Let $\{\omega_k\}_k$ be a martingale difference sequence in a Hilbert space. Suppose that almost surely $\|\omega_k\| \le B$ and $\sum_{k=1}^t \mathbb{E}[\|\omega_k\|^2 | \omega_1, \ldots, \omega_{k-1}] \le L_t^2$. Then for any $0 < \delta < 1$, the following holds with probability at least $1 - \delta$,*

$$\sup_{1 \le j \le t} \left\| \sum_{k=1}^j \omega_k \right\| \le 2 \left(\frac{B}{3} + L_t \right) \log \frac{2}{\delta}.$$

The required bounds B and L_t will be presented in subsections 5.3 and 5.4, respectively.

5.2 Initial Error

Lemma 13 *Let $\eta_k = \eta_1 k^{-\theta}$ with $\theta \in (0,1]$ and $\eta_1 \in (0,1)$. Then*

$$\|\Pi_1^t r_1\| \leq \begin{cases} C_0 t^{-\theta}, & \text{when } \theta < 1, \\ C_0 t^{-\lambda_r \eta_1}, & \text{when } \theta = 1, \end{cases}$$

where

$$C_0 = \begin{cases} \|r_1\| \exp\left\{\frac{\lambda_r \eta_1}{1-\theta}\right\} \left(\frac{\theta}{\lambda_r \eta_1 \mathrm{e}}\right)^{\frac{\theta}{1-\theta}}, & \text{when } \theta < 1, \\ \|r_1\|, & \text{when } \theta = 1. \end{cases}$$

Proof By our choice of x^*, we know that r_1 belongs to the subspace $V_0^\perp$. Thus, we have

$$\|\Pi_1^t r_1\| \leq \|\Pi_1^t|_{V_0^\perp}\| \|r_1\|.$$

Here $\Pi_1^t|_{V_0^\perp}$ denotes the restriction of the self adjoint operator Π_1^t onto $V_0^\perp$. Since $\{\lambda_l : l = 1, 2, \cdots, r\}$ are the eigenvalues of C_{ρ_X} restricted to $V_0^\perp$, $\lambda_1 \leq 1$ and $\eta_1 < 1$, we have

$$\|\Pi_1^t|_{V_0^\perp}\| = \sup_{1 \leq l \leq r} \prod_{k=1}^t (1 - \eta_k \lambda_l) \leq \prod_{k=1}^t (1 - \eta_1 \lambda_r k^{-\theta}) \leq \exp\left\{-\lambda_r \eta_1 \sum_{k=1}^t k^{-\theta}\right\}.$$

Applying part (d) of Lemma 10, we get our desired result. ■

5.3 Bounding the Residual Sequence

To bound $\omega_k = \eta_k \Pi_{k+1}^t \chi_k$, we start with a rough bound for $\|r_t\|$.

Lemma 14 *Assume that for some constant $M > 0$, $|\widetilde{y}| \leq M$ almost surely. Let $\theta \in [0,1]$ and $\eta_t = \eta_1 t^{-\theta}$ with $\eta_1 \in (0,1)$. Then for any $t \in \mathbb{N}$, we have almost surely*

$$\|r_t\| \leq \begin{cases} C_1 t^{\frac{1-\theta}{2}}, & \text{when } \theta \in [0,1), \\ C_1 \sqrt{\log(et)}, & \text{when } \theta = 1, \end{cases} \tag{32}$$

where C_1 is a constant independent of t given by

$$C_1 = \begin{cases} \sqrt{\frac{\|r_1\|^2 + \eta_1 (M + \|x^*\|)^2}{1-\theta}}, & \text{when } \theta \in [0,1), \\ \sqrt{\|r_1\|^2 + \eta_1 (M + \|x^*\|)^2}, & \text{when } \theta = 1. \end{cases}$$

Proof Rewrite (19) with $x_t = x^* + r_t$ as

$$\begin{aligned} \|r_{t+1}\|^2 &= \|r_t\|^2 + 2\eta_t(\tilde{y}_t - \langle \psi_t, x^* \rangle - \langle \psi_t, r_t \rangle)\langle \psi_t, r_t \rangle \\ &\quad + \eta_t^2 (\tilde{y}_t - \langle \psi_t, x^* \rangle - \langle \psi_t, r_t \rangle)^2 \\ &= \|r_t\|^2 + \mathcal{F}(\langle \psi_t, r_t \rangle), \end{aligned}$$

where $\mathcal{F}:\mathbb{R}\to\mathbb{R}$ is a quadratic function given by

$$\mathcal{F}(\mu)=\mathcal{F}_{\eta_t,\tilde{y}_t,\psi_t,x^*}(\mu)=\eta_t(\eta_t-2)\mu^2+2\eta_t(1-\eta_t)(\tilde{y}_t-\langle\psi_t,x^*\rangle)\mu+\eta_t^2(\tilde{y}_t-\langle\psi_t,x^*\rangle)^2.$$

Note that $\eta_t(\eta_t-2)\le 0$ by $0<\eta_t\le\eta_1\le 1$. A simple calculation shows that

$$\max_{x\in\mathbb{R}}\mathcal{F}(x)=-\frac{\eta_t^2(1-\eta_t)^2(\tilde{y}_t-\langle\psi_t,x^*\rangle)^2}{\eta_t(\eta_t-2)}+\eta_t^2(\tilde{y}_t-\langle\psi_t,x^*\rangle)^2=\frac{\eta_t(\tilde{y}_t-\langle\psi_t,x^*\rangle)^2}{2-\eta_t}.$$

Since $|\tilde{y}_t|\le M$ almost surely and $\|\psi_t\|=1$,

$$|\tilde{y}_t-\langle\psi_t,x^*\rangle|\le|\tilde{y}_t|+\|\psi_t\|\|x^*\|\le M+\|x^*\|.$$

Thus,

$$\|r_{t+1}\|^2\le\|r_t\|^2+\frac{\eta_t(\tilde{y}_t-\langle\psi_t,x^*\rangle)^2}{2-\eta_t}\le\|r_t\|^2+\eta_t(M+\|x^*\|)^2.$$

Using this relationship iteratively yields

$$\|r_{t+1}\|^2\le\|r_1\|^2+\sum_{k=1}^{t}\eta_k(M+\|x^*\|)^2=\|r_1\|^2+\eta_1(M+\|x^*\|)^2\sum_{k=1}^{t}k^{-\theta}.$$

Since that

$$\sum_{k=1}^{t}k^{-\theta}\le 1+\sum_{k=2}^{t}\int_{k-1}^{k}x^{-\theta}dx=\begin{cases}\frac{t^{1-\theta}-\theta}{1-\theta}, & \text{when } \theta\in[0,1),\\ \log(et), & \text{when } \theta=1,\end{cases}$$

we get

$$\|r_t\|^2\le\begin{cases}\frac{\|r_1\|^2+\eta_1(M+\|x^*\|)^2}{1-\theta}t^{1-\theta}, & \text{when } \theta\in[0,1),\\ (\|r_1\|^2+\eta_1(M+\|x^*\|)^2)\log(et), & \text{when } \theta=1,\end{cases}$$

which leads to the desired result. ■

5.4 Estimating Conditional Variance and Upper Bound

In this subsection, we give bounds for the two terms $\sum_{k=1}^{t}\eta_k^2\mathbb{E}[\|\Pi_{k+1}^{t}\chi_k\|^2|z_1,\dots,z_{k-1}]$ and $\sup_{1\le k\le t}\|\eta_k\Pi_{k+1}^{t}\chi_k\|$ required in applying the Pinelis-Bernstein inequality.

Lemma 15 *Let $\eta_k=\eta_1k^{-\theta}$ with $\theta\in(0,1]$ and $\eta_1\in(0,1)$. Then almost surely we have*

$$\begin{aligned}&\sum_{k=1}^{t}\eta_k^2\mathbb{E}[\|\Pi_{k+1}^{t}\chi_k\|^2|z_1,\dots,z_{k-1}]\\ \le&\sum_{k=1}^{t}\eta_1^2k^{-2\theta}\exp\left\{-2\eta_1\lambda_r\sum_{j=k+1}^{t}j^{-\theta}\right\}\left(\mathcal{E}(f_{\mathcal{H}})+\|r_k\|_2^2\right).\end{aligned}\tag{33}$$

Proof Recall that both ψ_k and r_k belong to $V_0^\perp$ almost surely for each $k \in \mathbb{N}$. As a result, χ_k also belongs to $V_0^\perp$ almost surely for each k. Hence

$$\sum_{k=1}^t \eta_k^2 \mathbb{E}[\|\Pi_{k+1}^t \chi_k\|^2 | z_1, \ldots, z_{k-1}] \leq \sum_{k=1}^t \eta_k^2 \|\Pi_{k+1}^t|_{V_0^\perp}\|^2 \mathbb{E}[\|\chi_k\|^2 | z_1, \ldots, z_{k-1}].$$

Since $\eta_1 < 1$ and $\lambda_1 \leq 1$, we have

$$\begin{aligned} \left\|\Pi_{k+1}^t|_{V_0^\perp}\right\| &= \sup_{1\leq l\leq r} \prod_{j=k+1}^t (1-\eta_j\lambda_l) \leq \prod_{j=k+1}^t (1-\eta_j\lambda_r) \\ &\leq \exp\left\{-\lambda_r \sum_{j=k+1}^t \eta_j\right\} = \exp\left\{-\lambda_r\eta_1 \sum_{j=k+1}^t j^{-\theta}\right\}. \end{aligned} \tag{34}$$

Thus,

$$\begin{aligned} &\sum_{k=1}^t \eta_k^2 \mathbb{E}[\|\Pi_{k+1}^t \chi_k\|^2 | z_1, \ldots, z_{k-1}] \\ &\leq \sum_{k=1}^t \eta_k^2 \exp\left\{-2\lambda_r\eta_1 \sum_{j=k+1}^t j^{-\theta}\right\} \mathbb{E}[\|\chi_k\|^2 | z_1, \ldots, z_{k-1}]. \end{aligned} \tag{35}$$

Since r_k does not depend on z_k, we see from $\|\psi_k\| = 1$, $\mathbb{E}[\tilde{y}_k|\psi_k] = f_\rho(\psi_k)$ and (17) that

$$\begin{aligned} &\mathbb{E}_{z_k}[\langle(\tilde{y}_k - \langle\psi_k, x^*\rangle)\psi_k, (C_{\rho_X} - \psi_k\psi_k^T)r_k\rangle] \\ &= \mathbb{E}_{z_k}[(\tilde{y}_k - \langle\psi_k, x^*\rangle)\langle\psi_k, C_{\rho_X} r_k\rangle] - \mathbb{E}_{z_k}[(\tilde{y}_k - \langle\psi_k, x^*\rangle)\langle\psi_k, r_k\rangle\|\psi_k\|^2] \\ &= \mathbb{E}_{\psi_k}[(f_\rho(\psi_k) - \langle\psi_k, x^*\rangle)\langle\psi_k, C_{\rho_X} r_k\rangle] - \mathbb{E}_{\psi_k}[(f_\rho(\psi_k) - \langle\psi_k, x^*\rangle)\langle\psi_k, r_k\rangle] = 0. \end{aligned}$$

It thus follows that

$$\begin{aligned} &\mathbb{E}[\|\chi_k\|^2 | z_1, \ldots, z_{k-1}] = \mathbb{E}_{z_k}[\|\chi_k\|^2] \\ &= \mathbb{E}_{z_k}[(\tilde{y}_k - \langle\psi_k, x^*\rangle)^2] + \mathbb{E}_{z_k}[\|(C_{\rho_X} - \psi_k\psi_k^T)r_k\|^2] \\ &= \mathcal{E}(f_{\mathcal{H}}) + \mathbb{E}_{z_k}\langle(C_{\rho_X} - C_{\rho_X}^2)r_k, r_k\rangle \\ &\leq \mathcal{E}(f_{\mathcal{H}}) + \|r_k\|_2^2. \end{aligned}$$

Putting the above bound into (35), we get the desire result. ■

Lemma 16 *Assume that for some constant $M > 0$, $|\tilde{y}| \leq M$ almost surely. Let $\theta \in [0, 1]$ and $\eta_t = \eta_1 t^{-\theta}$ with $\eta_1 \in (0, 1)$. Then for any $t \in \mathbb{N}$, we have almost surely*

$$\sup_{1\leq k\leq t} \|\eta_k \Pi_{k+1}^t \chi_k\| \leq \begin{cases} C_2 t^{-\theta} \max\left\{\sup_{1\leq k\leq t}\|r_k\|,\ 1\right\}, & \text{when } \theta < 1, \\ C_2 t^{-\lambda_r\eta_1} \max\left\{\sup_{1\leq k\leq t}\|r_k\|,\ 1\right\}, & \text{when } \theta = 1, \end{cases} \tag{36}$$

where C_2 is a constant given by

$$C_2 = \begin{cases} \eta_1(M + \|x^*\| + 2)\left(2^\theta + \left(\frac{\theta}{\mathrm{e}\lambda_r\eta_1(1-2^{\theta-1})}\right)^{\frac{\theta}{1-\theta}}\right), & \text{when } \theta < 1, \\ \eta_1(M + \|x^*\| + 2)2^{\lambda_r\eta_1}, & \text{when } \theta = 1. \end{cases}$$

Proof Let $k \in \{1, \ldots, t\}$. From the definition of χ_k, we have

$$\|\chi_k\| \leq (|\tilde{y}_k| + \|\psi_k\|\|x^*\|)\|\psi_k\| + \|C_{\rho_X} - \psi_k\psi_k^T\|\|r_k\|.$$

But $|\tilde{y}_k| \leq M$, $\|\psi_k\| = 1$ and $\|C_{\rho_X}\| \leq 1$. So we have

$$\|\chi_k\| \leq M + \|x^*\| + 2\|r_k\| \leq (M + \|x^*\| + 2)\max\{\|r_k\|, 1\}.$$

This together with (34) and the fact that χ_k belongs to $V_0^\perp$ implies

$$\begin{aligned}
\|\eta_k \Pi_{k+1}^t \chi_k\| &\leq \eta_1 k^{-\theta} \|\Pi_{k+1}^t|_{V_0^\perp}\| \|\chi_k\| \\
&\leq \eta_1 (M + \|x^*\| + 2) k^{-\theta} \|\Pi_{k+1}^t|_{V_0^\perp}\| \max\{\|r_k\|, 1\} \\
&\leq \eta_1 (M + \|x^*\| + 2) k^{-\theta} \exp\left\{-\lambda_r \eta_1 \sum_{j=k+1}^{t} j^{-\theta}\right\} \max\{\|r_k\|, 1\}.
\end{aligned}$$

What is left is to estimate

$$I_k := k^{-\theta} \exp\left\{-\lambda_r \eta_1 \sum_{j=k+1}^{t} j^{-\theta}\right\}.$$

For $\theta \in [1/2, 1)$, applying part (c) of Lemma 10 gives

$$I_k \leq k^{-\theta} \exp\left\{-\frac{\lambda_r \eta_1}{1-\theta}[(t+1)^{1-\theta} - (k+1)^{1-\theta}]\right\}.$$

If $k \geq t/2$, then $k^{-\theta} \leq 2^\theta t^{-\theta}$ and thus

$$I_k \leq 2^\theta t^{-\theta}.$$

If $1 \leq k < t/2$, then we have $k+1 \leq (t+1)/2$ and $(t+1)^{1-\theta} - (k+1)^{1-\theta} \geq (1-2^{\theta-1})(t+1)^{1-\theta}$. It follows that

$$I_k \leq \exp\left\{-\frac{\lambda_r \eta_1 (1 - 2^{\theta-1})}{1-\theta} t^{1-\theta}\right\}.$$

Applying part (a) of Lemma 10 with $x = t^{1-\theta}$, $\nu = \frac{\lambda_r \eta_1 (1-2^{\theta-1})}{1-\theta}$ and $a = \frac{\theta}{1-\theta}$, we get

$$I_k \leq \left(\frac{\theta}{\mathrm{e}\lambda_r \eta_1 (1 - 2^{\theta-1})}\right)^{\frac{\theta}{1-\theta}} t^{-\theta}.$$

For $\theta = 1$, by part (c) of Lemma 10, with $\lambda_r \eta_1 < 1$, we have

$$I_k \leq k^{-1} \left(\frac{t+1}{k+1}\right)^{-\lambda_r \eta_1} = \left(\frac{t}{t+1} \cdot \frac{k+1}{k}\right)^{\lambda_r \eta_1} t^{-\lambda_r \eta_1} k^{\lambda_r \eta_1 - 1} \leq 2^{\lambda_r \eta_1} t^{-\lambda_r \eta_1}.$$

From the above analysis, we conclude the desired result. ■

5.5 Preliminary Error Analysis

Based on the above estimates, we can apply Lemma 12 to obtain an error bound.

Proposition 17 *Under the assumptions of Theorem 5, for some $x^* \in \mathbb{R}^d$ and for any $0 < \delta < 1$ and fixed $t \in \mathbb{N}$, with confidence at least $1-\delta$, we have*

$$\|x_{t+1} - x^*\| \leq \begin{cases} \widetilde{C}_2 t^{\frac{1}{2}-\theta} \log \frac{2}{\delta}, & \text{when } \theta \in [\frac{1}{3}, 1), \\ \widetilde{C}_2 t^{-\lambda_r \eta_1} \sqrt{\log(\mathrm{e}t)} \log \frac{2}{\delta}, & \text{when } \theta = 1, \end{cases} \tag{37}$$

where $\widetilde{C}_2$ is a positive constant independent of t or δ (given explicitly in the proof).

Proof To apply the Pinelis-Bernstein inequality to estimate $\| \sum_{k=1}^t \eta_k \Pi_{k+1}^t \chi_k \|$, we need bounds B and L_t.

By Lemmas 14 and 16, we have

$$\sup_{1 \leq k \leq t} \|\eta_k \Pi_{k+1}^t \chi_k\| \leq \begin{cases} C_2(C_1+1) t^{\frac{1-3\theta}{2}}, & \text{when } \theta < 1, \\ C_2(C_1+1) t^{-\lambda_r \eta_1} \sqrt{\log(\mathrm{e}t)}, & \text{when } \theta = 1. \end{cases} \tag{38}$$

By Lemmas 15 and 14, we get

$$\begin{aligned} &\sum_{k=1}^t \eta_k^2 \mathbb{E}[\|\Pi_{k+1}^t \chi_k\|^2 | z_1, \ldots, z_{k-1}] \\ &\leq \begin{cases} C_3 \sum_{k=1}^t k^{-(3\theta-1)} \exp\left\{-2\lambda_r \eta_1 \sum_{j=k+1}^t j^{-\theta}\right\}, & \text{when } \theta \in [\frac{1}{3}, 1), \\ C_3 \log(et) \sum_{k=1}^t k^{-2} \exp\left\{-2\lambda_r \eta_1 \sum_{j=k+1}^t j^{-1}\right\}, & \text{when } \theta = 1, \end{cases} \end{aligned}$$

where

$$C_3 = (\mathcal{E}(f_{\mathcal{H}}) + C_1^2)\eta_1^2.$$

Applying part (b) of Lemma 10 with $\nu = 2\lambda_r \eta_1 < 1$, $q_1 = \theta$ and $q_2 = 3\theta - 1$, we have for $\theta < 1$,

$$\begin{aligned} &\sum_{k=1}^t k^{-(3\theta-1)} \exp\left\{-2\lambda_r \eta_1 \sum_{j=k+1}^t j^{-\theta}\right\} \\ &\leq \left(\frac{2^{4\theta-1}}{2\lambda_r \eta_1} + \left(\frac{3\theta}{2\lambda_r \eta_1 \mathrm{e}(1-2^{\theta-1})} \right)^{\frac{3\theta}{1-\theta}} \right) t^{1-2\theta} + t^{1-3\theta}, \end{aligned}$$

and for $\theta = 1$,

$$\sum_{k=1}^t k^{-2} \exp\left\{-2\lambda_r \eta_1 \sum_{j=k+1}^t j^{-1}\right\} \leq \frac{4}{1-2\lambda_r \eta_1} t^{-2\lambda_r \eta_1} + t^{-2}.$$

Therefore, we get

$$\sum_{k=1}^t \eta_k^2 \mathbb{E}[\|\Pi_{k+1}^t \chi_k\|^2 | z_1, \ldots, z_{k-1}] \leq \begin{cases} C_4 t^{1-2\theta}, & \text{when } \theta \in [\frac{1}{3}, 1), \\ C_4 t^{-2\lambda_r \eta_1} \log(et), & \text{when } \theta = 1, \end{cases} \tag{39}$$

with

$$C_4 = \begin{cases} C_3\left(\frac{2^{4\theta-1}}{2\lambda_r\eta_1} + \left(\frac{3\theta}{2\lambda_r\eta_1 \mathrm{e}(1-2^{\theta-1})}\right)^{\frac{3\theta}{1-\theta}} + 1\right), & \text{when } \theta \in [\frac{1}{3}, 1), \\ C_3\frac{5-2\lambda_r\eta_1}{1-2\lambda_r\eta_1}, & \text{when } \theta = 1. \end{cases}$$

Applying Lemma 12 to the martingale difference sequence $\{\omega_k := \eta_k \Pi_{k+1}^t \chi_k\}_k$ with B and L_t given by (38) and (39) respectively, we know that with probability at least $1-\delta$,

$$\sup_{1\le j\le t}\left\|\sum_{k=1}^{j}\eta_k\Pi_{k+1}^t\chi_k\right\| \le \begin{cases} C_5 t^{\frac{1-2\theta}{2}}\log\frac{2}{\delta}, & \text{when } \theta \in [\frac{1}{3}, 1), \\ C_5 t^{-\lambda_r\eta_1}\sqrt{\log(\mathrm{e}t)}\log\frac{2}{\delta}, & \text{when } \theta = 1, \end{cases}$$

where

$$C_5 = 2\left(C_2(C_1+1)/3 + \sqrt{C_4}\right).$$

Putting this bound into (31) with t replaced by j, and then applying Lemma 13 to bound the initial error, we get the desired result with $\widetilde{C}_2 = C_0 + C_5$ from Lemma 12. ■

In the above procedure, we have used a rough bound (32) for $\|r_t\|$. This rough bound tends to ∞ as t becomes large. In contrast, the bound provided in Proposition 17 tends to 0 (when $\theta \in (1/2, 1]$) and is much better. But this bound holds with confidence. We shall use this refined bound to improve our estimates in the following subsection.

5.6 Improved Error Analysis

In this subsection, we prove our third main result by improving the preliminary confidence-based error bound in Proposition 17.

Proof of Theorem 5 When $\theta = 1$, our desired bound follows from (37) with $\widetilde{C}_1 = 2\widetilde{C}_2$.

It remains to prove the case $\theta \in [1/2, 1)$. Let $T \in N$. Applying Proposition 17 with $t = 1, \cdots, T$, and taking the union event followed by rescaling, we know that there exists a subset Z_δ^T of Z^T with measure at least $1-\delta$ such that

$$\|r_t\| \le C_6 \log\frac{2}{\delta}\log T, \qquad \forall\, t = 1, \ldots, T+1,\ (z_1, \ldots, z_T) \in Z_\delta^T, \tag{40}$$

where $C_6 = 2\widetilde{C}_2 + \|r_1\|$.

Now we turn to the essential part of the proof. Define another martingale difference sequence $\{\widetilde{\omega}_k\}_k$ by multiplying the one in the proof of Proposition 17 by a characteristic function $\mathbf{1}_{\{\|r_k\| \le C_6 \log\frac{2}{\delta}\log T\}}$ as

$$\widetilde{\omega}_k = \eta_k \Pi_{k+1}^T \chi_k \mathbf{1}_{\{\|r_k\| \le C_6 \log\frac{2}{\delta}\log T\}}.$$

From (36) and the multiplication with the characteristic function $\mathbf{1}_{\{\|r_k\| \le C_6 \log\frac{2}{\delta}\log T\}}$, we have

$$\sup_{1\le k\le T}\|\widetilde{\omega}_k\| \le C_2 C_6 \log\left(\frac{2}{\delta}\right)(\log T) T^{-\theta}. \tag{41}$$

Notice that the characteristic function $\mathbf{1}_{\{\|r_k\|\le C_6\log\frac{2}{\delta}\log T\}}$ is independent of z_k. Also, from the proof of Lemma 15, we know that for each $k\in\{1,\dots,T\}$,

$$\mathbb{E}[\|\Pi_{k+1}^t\chi_k\|^2|z_1,\dots,z_{k-1}]\le\exp\left\{-\lambda_r\eta_1\sum_{j=k+1}^t j^{-\theta}\right\}\left(\mathcal{E}(f_{\mathcal{H}})+\|r_k\|_2^2\right).$$

It follows by setting $C_7=(\mathcal{E}(f_{\mathcal{H}})+C_6^2)\eta_1^2$ that

$$\sum_{k=1}^T\mathbb{E}\left[\|\widetilde{\omega}_k\|^2|z_1,\dots,z_{k-1}\right]\le C_7\left(\log\frac{2}{\delta}\log T\right)^2\sum_{k=1}^T k^{-2\theta}\exp\left\{-2\lambda_r\eta_1\sum_{j=k+1}^T j^{-\theta}\right\}.$$

Applying part (b) of Lemma 10 yields

$$\begin{aligned}&\sum_{k=1}^T\mathbb{E}\left[\|\widetilde{\omega}_k\|^2|z_1,\dots,z_{k-1}\right]\\&\le C_7\left(\log\frac{2}{\delta}\log T\right)^2\left(\frac{2^{3\theta}}{2\lambda_r\eta_1}+\left(\frac{1+2\theta}{2\lambda_r\eta_1\mathrm{e}(1-2^{\theta-1})}\right)^{\frac{1+2\theta}{1-\theta}}+1\right)T^{-\theta}.\end{aligned}$$

Using this bound as L_T and (41) as the bound B in Lemma 12, we know that there exists another subset $\tilde{Z}_\delta^T$ of Z^T with measure at least $1-\delta$ such that for every $(z_1,\dots,z_T)\in\tilde{Z}_\delta^T$, there holds

$$\left\|\sum_{k=1}^T\widetilde{\omega}_k\right\|\le C_8T^{\frac{-\theta}{2}}\left(\log\frac{2}{\delta}\right)^2\log T,$$

where

$$C_8=\frac{2C_2C_6}{3}+2\sqrt{C_7}\left(\frac{2^{3\theta}+2\lambda_r\eta_1}{2\lambda_r\eta_1}+\left(\frac{1+2\theta}{2\lambda_r\eta_1\mathrm{e}(1-2^{\theta-1})}\right)^{\frac{1+2\theta}{1-\theta}}\right)^{\frac{1}{2}}.$$

This together with (40) tells us that for every $(z_1,\dots,z_T)\in Z_\delta^T\cap\tilde{Z}_\delta^T$, there holds

$$\left\|\sum_{k=1}^T\eta_k\Pi_{k+1}^T\chi_k\right\|\le C_8T^{\frac{-\theta}{2}}\left(\log\frac{2}{\delta}\right)^2\log T.\tag{42}$$

The subset $Z_\delta^T\cap\tilde{Z}_\delta^T$ has measure at least $1-2\delta$. Therefore, we can put (42) into (31), and apply Lemma 13 to bound the initial error, which proves Theorem 5 for the case $\theta\in[1/2,1)$ after scaling δ to $\delta/2$ and setting the constant $\widetilde{C}_1=C_0+C_8$. ■

6. Almost Sure Convergence

In this section, we prove the almost sure convergence of the randomized Kaczmarz algorithm. Recall that the almost sure convergence of a sequence of random variables $\{X_n\}$ towards X means that

$$\mathbb{P}\left(\lim_{n\to\infty}X_n=X\right)=1,$$

or equivalently,

$$\lim_{n\to\infty} \mathbb{P}\Big(\sup_{k\geq n} |X_k - X| > \varepsilon \Big) = 0 \quad \text{for any } \varepsilon > 0.$$

The Borel-Cantelli Lemma (see e.g. (Klenke, 2010)) asserts for a sequence $(E_n)_n$ of events that if the sum of the probabilities is finite $\sum_{n=1}^{\infty} \mathbb{P}(E_n) < \infty$, then the probability that infinitely many of them occur is 0, that is, $\mathbb{P}\left(\limsup_{n\to\infty} E_n\right) = \mathbb{P}\left(\cap_{n=1}^{\infty} \cup_{k=n}^{\infty} E_n\right) = 0$. The following lemma is an easy consequence of the Borel-Cantelli Lemma. We give the proof for completeness.

Lemma 18 *Let $\{X_n\}$ be a sequence of events in some probability space and $\{\varepsilon_n\}$ be a sequence of positive numbers satisfying $\lim_{n\to\infty} \varepsilon_n = 0$. If*

$$\sum_{n=1}^{\infty} \mathbb{P}\left(|X_n - X| > \varepsilon_n\right) < \infty,$$

then X_n converges to X almost surely.

Proof Since $\lim_{n\to\infty} \varepsilon_n = 0$, for any $\varepsilon > 0$, there exists some $n \in \mathbb{N}$ such that for all $k \geq n$, $\varepsilon_k < \varepsilon$. Thus,

$$\mathbb{P}\Big(\sup_{k\geq n} |X_k - X| > \varepsilon \Big) \leq \mathbb{P}\Big(\bigcup_{k\geq n} (|X_k - X| > \varepsilon_k) \Big) \leq \sum_{k\geq n} \mathbb{P}\Big(|X_k - X| > \varepsilon_k \Big).$$

Letting $n \to \infty$, one gets $\mathbb{P}\left(\sup_{k\geq n} |X_k - X| > \varepsilon\right) \to 0$. This proves the result. ■

Now we can apply Lemma 18 to prove our last main result.

Proof of Theorem 6 Set

$$\Lambda_t = \begin{cases} t^{-\theta/2} & \text{when } \theta < 1, \\ t^{-\lambda_r \eta_1} & \text{when } \theta = 1. \end{cases}$$

By Theorem 5, we have for any $t \geq 2$ and $0 < \delta_t < 1$,

$$\mathbb{P}\left(\Lambda_t^{\epsilon-1} \|x_{t+1} - x^*\| > \widetilde{C}_1 \Lambda_t^{\epsilon} \left(\log \frac{4}{\delta_t} \right)^2 \log t \right) \leq \delta_t.$$

Choose $\delta_t = t^{-2}$, and $\varepsilon_t = \widetilde{C}_1 \Lambda_t^{\epsilon} \left(\log 4/\delta_t\right)^2 \log t$. Obviously

$$\sum_{t=2}^{\infty} \mathbb{P}\left(\Lambda_t^{\epsilon-1} \|x_{t+1} - x^*\| > \varepsilon_t \right) \leq \sum_{t=2}^{\infty} \delta_t < \infty$$

and

$$\varepsilon_t \leq 4\widetilde{C}_1 \Lambda_t^{\epsilon} \log^3(2t) \to 0, \quad \text{as } t \to \infty.$$

Then our conclusion of Theorem 6 follows from Lemma 18. ■

Remark 19 *The above method of proof can be used to get a more quantitative estimate for the almost sure convergence of the Kaczmarz algorithm with noiseless random measurements (Chen and Powell, 2012). In that setting,* $\eta_t \equiv 1$, $y_t = f_\rho(\psi_t)$ *and* $r = d$. *It was shown in (Strohmer and Vershynin, 2009; Chen and Powell, 2012) that with* $q = 1 - \lambda_r$,

$$\mathbb{E}\|x_{t+1} - x^*\|^2 \le q^t \|r_1\|^2.$$

It follows from the Chebyshev inequality that for any $\epsilon \in (0, 1)$,

$$\mathbb{P}\left(q^{t(\epsilon-1)}\|x_{t+1} - x^*\|^2 > q^{t\epsilon}t^2\right) = \mathbb{P}\left(\|x_{t+1} - x^*\|^2 > q^t t^2\right) \le \frac{\mathbb{E}[\|x_{t+1} - x^*\|^2]}{q^t t^2}.$$

Thus, we get

$$\mathbb{P}\left(q^{t(\epsilon-1)}\|x_{t+1} - x^*\|^2 > q^{t\epsilon}t^2\right) \le \|r_1\|t^{-2}.$$

Obviously, $q^{t\epsilon}t^2 \to 0$ *as* $t \to \infty$, *and* $\sum_{t=1}^{\infty} \|r_1\|t^{-2} < \infty$. *Applying Lemma 18 with* $\varepsilon_t = q^{t\epsilon}t^2$, *we know that for any* $\epsilon \in (0, 1)$,

$$\lim_{t\to\infty} (1 - \lambda_r)^{t(\epsilon-1)}\|x_{t+1} - x^*\|^2 = 0 \quad \textit{almost surely.}$$

7. Simulations and Discussions

In this section we provide some numerical simulations and further discussions on our error analysis.

To illustrate our derived convergence rates and compare with the existing literature, we carry out numerical simulations corresponding to Example 2 with the same data distributions as in (Needell, 2010): $m = 200, d = 100$, $A \in \mathbb{R}^{200\times100}$ is a Gaussian matrix with each entry drawn independently from the standard normal distribution $N(0, 1)$, and $y \in \mathbb{R}^{100}$ is a Gaussian noise with each component drawn independently from the normal distribution with mean 0 and standard deviation 0.02. The measurement vectors $\{\psi_t = \frac{1}{\|\varphi_t\|}\varphi_t\}$ are drawn from the normalized rows of A as in Example 2 and $\{\widetilde{y}_t = y_t/\|\varphi_t\|\}$ with mean $x^* = 0$. We conduct 100 trials for each choice of the relaxation parameter sequences $\eta_t = 1, \eta_t = 1/\sqrt{t}, \eta_t = 1/t$. In each trial, algorithm (4) is run 100 times with random Gaussian initial vectors of norm $\|x_1\| = 0.02$. Figure 1 depicts the error $\|x_{t+1} - x^*\|$ for $t = 1, \ldots, 1500$ (averaged with 100 trials and 100 initial vectors). The black line is a plot with the constant relaxation parameter sequence $\eta_t = 1$, which verifies the divergence of the algorithm, as proved in (Needell, 2010). The blue line is a plot with $\eta_t = 1/\sqrt{t}$, which hints a slow convergence of the algorithm. The red line is a plot with $\eta_t = 1/t$, which confirms a faster convergence. The above simulations are consistent with our error analysis.

In this paper, a learning theory approach to the relaxed randomized Kaczmarz algorithm is presented. It yields new results and observations including a necessary and sufficient condition (9), stated in Theorem 3, for the convergence in expectation when the sampling process is noisy or nonlinear. For noise-free and linear sampling processes (that is, $\mathcal{E}(f_{\mathcal{H}}) = 0$), we can see from Remark 11 with $\eta_t \equiv 1$ that $\mathbb{E}_{z_1,\ldots,z_T}[\|x_{T+1} - x^*\|^2] \le \|x_1 - x^*\|^2(1 - \lambda_r)^T$. This exponential convergence result was proved in (Strohmer and Vershynin, 2009) for Example 2 under the restriction that the matrix A has full column rank, where the number

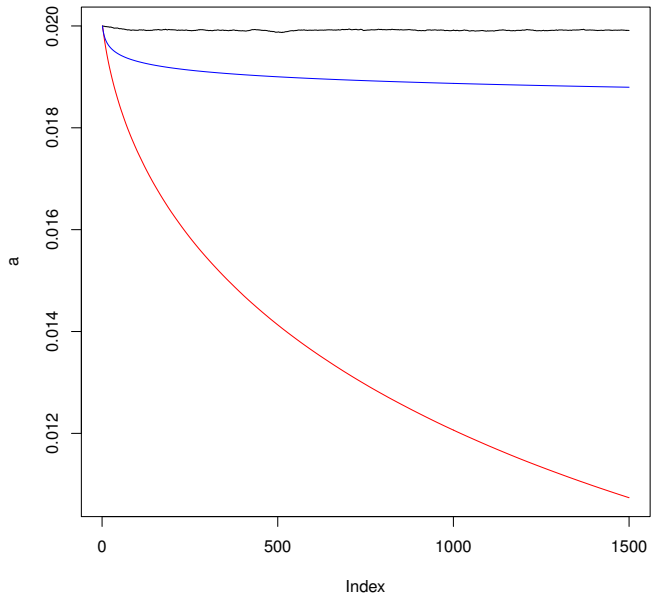


Figure 1: Error of the relaxed randomized Kaczmarz algorithm with $\eta_t = 1$ (black line), $\eta_t = 1/\sqrt{t}$ (blue line), and $\eta_t = 1/t$ (red line)

$1-\lambda_r$ is replaced by a quantity involving $\|A^{-1}\| = \inf\{M : M\|Ax\| \geq \|x\| \text{ for all } x\}$. Our result is more general (valid for underdetermined systems with $\|A^{-1}\| = \infty$).

In the framework of Kaczmarz algorithms, we consider online learning algorithms associated with the least squares loss. It would be interesting to extend our study to algorithms associated with more general loss functions (Ying and Zhou, 2006) such as hinge loss, and to consider error analysis without requiring the approximation error (Ying and Zhou, 2006) tending to zero.

Acknowledgments

The work described in this paper is supported partially by the Research Grants Council of Hong Kong [Project No. CityU 104113]. The authors would like to thank the referees for constructive suggestions and Dr. Xin Guo for helping with the numerical simulations. The corresponding author is Ding-Xuan Zhou.

References

X. Chen and A. Powell. Almost sure convergence for the Kaczmarz algorithm with random measurements. *Journal of Fourier Analysis and Applications*, 18:1195–1214, 2012.

T. Hu, J. Fan, Q. Wu, and D. X. Zhou. Regularization schemes for minimum error entropy principle. *Analysis and Applications*, 13:437–455, 2015.

J. Johnston. *Econometric Methods*. McGraw Hill, New York, 1963.

S. Kaczmarz. Angenäherte auflösung von systemen linearer gleichungen. *Bulletin International de l'Académie Polonaise des Sciences et des Lettres A*, 35:355–357, 1937.

A. Klenke. *Probability Theory: A Comprehensive Course*. Springer-Verlag, London, 2008.

D. Needell. Randomized Kaczmarz solver for noisy linear systems. *BIT. Numerical Mathematics*, 50:395–403, 2010.

I. Pinelis. Optimum bounds for the distributions of martingales in banach spaces. *Annals of Probability*, 22:1679–1706, 1994.

Y. Ying and M. Pontil. Online gradient descent learning algorithms. *Foundations of Computational Mathematics*, 8:561–596, 2008.

S. Smale and Y. Yao. Online learning algorithms. *Foundations of Computational Mathematics*, 6:145–170, 2005.

S. Smale and D. X. Zhou. Online learning with Markov sampling. *Analysis and Applications*, 7:87–113, 2009.

T. Strohmer and R. Vershynin. A randomized Kaczmarz algorithm with exponential convergence. *Journal of Fourier Analysis and Applications*, 15:262–278, 2009.

P. Tarrés and Y. Yao. Online learning as stochastic approximations of regularization paths. *IEEE Transactions on Information Theory*, 60:5716–5735, 2014.

V. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, 1998.

Y. Ying and D. X. Zhou. Online regularized classification algorithms. *IEEE Transactions on Information Theory*, 52:4775–4788, 2006.

A. Zouzias and N. M. Freris. Randomized extended Kaczmarz for solving least squares. *SIAM Journal on Matrix Analysis and Applications*, 34:773–793, 2013.

Journal of Machine Learning Research 16 (2015) 3367-3402 Submitted 10/14; Revised 2/15; Published 12/15

Matrix Completion and Low-Rank SVD via Fast Alternating Least Squares

Trevor Hastie HASTIE@STANFORD.EDU
Department of Statistics
Stanford University, CA 94305, USA

Rahul Mazumder RM3184@COLUMBIA.EDU
Department of Statistics
Columbia University
New York, NY 10027, USA

Jason D. Lee JDL17@STANFORD.EDU
Institute for Computational and Mathematical Engineering
Stanford University, CA 94305, USA

Reza Zadeh REZAB@STANFORD.EDU
Databricks
2030 Addison Street, Suite 610
Berkeley, CA 94704, USA

Editor: Guy Lebanon

Abstract

The matrix-completion problem has attracted a lot of attention, largely as a result of the celebrated Netflix competition. Two popular approaches for solving the problem are nuclear-norm-regularized matrix approximation (Candès and Tao, 2009; Mazumder et al., 2010), and maximum-margin matrix factorization (Srebro et al., 2005). These two procedures are in some cases solving equivalent problems, but with quite different algorithms. In this article we bring the two approaches together, leading to an efficient algorithm for large matrix factorization and completion that outperforms both of these. We develop a software package `softImpute` in R for implementing our approaches, and a distributed version for very large matrices using the `Spark` cluster programming environment

Keywords: matrix completion, alternating least squares, svd, nuclear norm

1. Introduction

We have an $m \times n$ matrix X with observed entries indexed by the set Ω; i.e. $\Omega = \{(i, j) : X_{ij} \text{ is observed}\}$. Following Candès and Tao (2009) we define the projection $P_\Omega(X)$ to be the $m \times n$ matrix with the observed elements of X preserved, and the missing entries replaced with 0. Likewise $P_\Omega^\perp$ projects onto the complement of the set Ω.

Inspired by Candès and Tao (2009), Mazumder et al. (2010) posed the following convex-optimization problem for completing X:

$$\underset{M}{\text{minimize}} \quad H(M) := \quad \tfrac{1}{2}\|P_\Omega(X - M)\|_F^2 + \lambda\|M\|_*, \tag{1}$$

where the *nuclear norm* $\|M\|_*$ is the sum of the singular values of M (a convex relaxation of the rank). They developed a simple iterative algorithm for solving Problem (1), with the following two steps iterated till convergence:

1. Replace the missing entries in X with the corresponding entries from the current estimate $\widehat{M}$:
$$\widehat{X} \leftarrow P_\Omega(X) + P_\Omega^\perp(\widehat{M}); \tag{2}$$

2. Update $\widehat{M}$ by computing the soft-thresholded SVD of $\widehat{X}$:
$$\begin{aligned} \widehat{X} &= UDV^T & (3)\\ \widehat{M} &\leftarrow U\mathcal{S}_\lambda(D)V^T, & (4) \end{aligned}$$
where the soft-thresholding operator $\mathcal{S}_\lambda$ operates element-wise on the diagonal matrix D, and replaces D_{ii} with $(D_{ii}-\lambda)_+$. With large λ many of the diagonal elements will be set to zero, leading to a low-rank solution for Problem (1).

For large matrices, step (3) could be a problematic bottleneck, since we need to compute the SVD of the filled matrix $\widehat{X}$. In fact, for the Netflix problem $(m,n) \approx (400K, 20K)$, which requires storage of 8×10^9 floating-point numbers (32Gb in single precision), which in itself could pose a problem. However, since only about 1% of the entries are observed (for the Netflix dataset), sparse-matrix representations can be used.

Mazumder et al. (2010) use two tricks to avoid these computational nightmares:

1. Anticipating a low-rank solution, they compute a reduced-rank SVD in step (3); if the smallest of the computed singular values is less than λ, this gives the desired solution. A reduced-rank SVD can be computed by using an iterative Lanczos-style method as implemented in PROPACK (Larsen, 2004), or by other alternating-subspace methods (Golub and Van Loan, 2012).

2. They rewrite $\widehat{X}$ in (2) as
$$\widehat{X} = \left[P_\Omega(X) - P_\Omega(\widehat{M})\right] + \widehat{M}; \tag{5}$$
The first piece is as sparse as X, and hence inexpensive to store and compute. The second piece is low rank, and also inexpensive to store. Furthermore, the iterative methods mentioned in step (1) require left and right multiplications of $\widehat{X}$ by *skinny* matrices, which can exploit this special structure.

This `softImpute` algorithm works very well, and although an SVD needs to be computed each time step (3) is evaluated, this step can use the previous solution as a warm start. As one gets closer to the solution, the warm starts tend to be better, and so the final iterations tend to be faster.

Mazumder et al. (2010) also considered a path of such solutions, with decreasing values of λ. As λ decreases, the rank of the solutions tend to increase, and at each λ_ℓ, the iterative algorithms can use the solution $\widehat{X}_{\lambda_{\ell-1}}$ (with $\lambda_{\ell-1} > \lambda_\ell$) as warm starts, padded with some additional dimensions.

Rennie and Srebro (2005) consider a different approach. They impose a rank constraint, and consider the problem

$$\underset{A,B}{\text{minimize}} \quad F(A,B) := \frac{1}{2}\|P_\Omega(X - AB^T)\|_F^2 + \frac{\lambda}{2}\left(\|A\|_F^2 + \|B\|_F^2\right), \tag{6}$$

where A is $m \times r$ and B is $n \times r$. This so-called maximum-margin matrix factorization (`MMMF`) criterion[1] is not convex in A and B, but it is bi-convex — for fixed B the function $F(A,B)$ is convex in A, and for fixed A the function $F(A,B)$ is convex in B. Alternating minimization algorithms (`ALS`) are often used to minimize Problem (6). Consider A fixed, and we wish to solve Problem (6) for B. It is easy to see that this problem decouples into n separate ridge regressions, with each column X_j of X as a response, and the r-columns of A as predictors. Since some of the elements of X_j are missing, and hence ignored, the corresponding rows of A are deleted for the jth regression. So these are really *separate* ridge regressions, in that the regression matrices are all different (even though they all derive from A). By symmetry, with B fixed, solving for A amounts to m separate ridge regressions.

There is a remarkable fact that ties the solutions to Problems (6) and (1) (Mazumder et al., 2010, for example). If the solution to Problem (1) has rank $q \leq r$, then it provides a solution to Problem (6). That solution is

$$\begin{aligned} \widehat{A} &= U_r \mathcal{S}_\lambda(D_r)^{\frac{1}{2}} \\ \widehat{B} &= V_r \mathcal{S}_\lambda(D_r)^{\frac{1}{2}}, \end{aligned} \tag{7}$$

where U_r, for example, represents the sub-matrix formed by the first r columns of U, and likewise D_r is the top $r \times r$ diagonal block of D. Note that for any solution to Problem (6), multiplying $\widehat{A}$ and $\widehat{B}$ on the right by an orthonormal $r \times r$ matrix R would be an equivalent solution. Likewise, any solution to Problem (6) with rank $r \geq q$ gives a solution to Problem (1).

In this paper we propose a new algorithm that profitably draws on ideas used both in `softImpute` and `MMMF`. Consider the two steps (3) and (4). We can alternatively solve the optimization problem

$$\underset{A,B}{\text{minimize}} \quad \frac{1}{2}\|\widehat{X} - AB^T\|_F^2 + \frac{\lambda}{2}(\|A\|_F^2 + \|B\|_B^2), \tag{8}$$

and as long as we use enough columns in A and B, we will have $\widehat{M} = \widehat{A}\widehat{B}^T$. There are several important advantages to this approach:

1. Since $\widehat{X}$ is fully observed, the (ridge) regression operator is the same for each column, and so is computed just once. This reduces the computation of an update of A or B over `ALS` by a factor of r.

2. By orthogonalizing the r-column matrices A or B at each iteration, the regressions are simply matrix multiplies, very similar to those used in the alternating subspace algorithms for computing the SVD.

1. Actually `MMMF` also refers to the margin-based loss function that they used, but we will nevertheless use this acronym.

3. This quadratic regularization amounts to shrinking the higher-order components more than the lower-order components, and this tends to offer a convergence advantage over the previous approach (compute the SVD, then soft-threshold).

4. Just like before, these operations can make use of the *sparse plus low-rank* property of $\widehat{X}$.

As an important additional modification, we replace $\widehat{X}$ at each step using the most recently computed $\widehat{A}$ or $\widehat{B}$. All combined, this hybrid algorithm tends to be faster than either approach on their own; see the simulation results in Section 6.1

For the remainder of the paper, we present this `softImpute-ALS` algorithm in more detail, and show that it convergences to the solution to Problem (1) for r sufficiently large. We demonstrate its superior performance on simulated and real examples, including the Netflix data. We briefly highlight two publicly available software implementations, and describe a simple approach to centering and scaling of both the rows and columns of the (incomplete) matrix.

2. Rank-restricted Soft SVD

In this section we consider a complete matrix X, and develop a new algorithm for finding a rank-restricted SVD. In the next section we will adapt this approach to the matrix-completion problem. We first give two theorems that are likely known to experts; the proofs are very short, so we provide them here for convenience.

Theorem 1 *Let $X_{m\times n}$ be a matrix (fully observed), and let $0 < r \le \min(m,n)$. Consider the optimization problem*

$$\underset{Z:\,\mathrm{rank}(Z)\le r}{\text{minimize}} \; F_\lambda(Z) := \tfrac{1}{2}||X-Z||_F^2 + \lambda||Z||_*. \tag{9}$$

A solution is given by

$$\widehat{Z} \;=\; U_r \mathcal{S}_\lambda(D_r) V_r^T, \tag{10}$$

where the rank-r SVD of X is $U_r D_r V_r^T$ and $\mathcal{S}_\lambda(D_r) = diag[(\sigma_1-\lambda)_+,\ldots,(\sigma_r-\lambda)_+]$.

Proof We will show that, for any Z the following inequality holds:

$$F_\lambda(Z) \ge f_\lambda(\boldsymbol{\sigma}(Z)) := \tfrac{1}{2}||\boldsymbol{\sigma}(X)-\boldsymbol{\sigma}(Z)||_2^2 + \lambda\sum_i \sigma_i(Z), \tag{11}$$

where, $f_\lambda(\boldsymbol{\sigma}(Z))$ is a function of the singular values of Z and $\boldsymbol{\sigma}(X)$ denotes the vector of singular values of X, such that $\sigma_i(X) \ge \sigma_{i+1}(X)$ for all $i = 1,\ldots,\min\{m,n\}$.

To show inequality (11) it suffices to show that:

$$\tfrac{1}{2}||X-Z||_F^2 \ge \tfrac{1}{2}||\boldsymbol{\sigma}(X)-\boldsymbol{\sigma}(Z)||_2^2,$$

which follows as an immediate consequence of the well-known Von-Neumann trace inequality (Mirsky, 1975; Stewart and Sun, 1990):

$$\mathrm{Tr}(X^TZ) \;:=\; \langle X,Z\rangle \;\le\; \sum_{i=1}^{\min\{m,n\}} \sigma_i(X)\sigma_i(Y),$$

that provides an upper bound to the trace of the product of two matrices in terms of the inner product of their singular values.

Observing that

$$\text{rank}(Z) \leq r \iff \|\boldsymbol{\sigma}(Z)\|_0 \leq r,$$

we have established:

$$\begin{aligned} &\min_{Z:\,\text{rank}(Z)\leq r} \left(\tfrac{1}{2}||X-Z||_F^2 + \lambda||Z||_*\right) \\ \geq &\min_{\boldsymbol{\sigma}(Z):\|\boldsymbol{\sigma}(Z)\|_0\leq r} \left(\tfrac{1}{2}||\boldsymbol{\sigma}(X)-\boldsymbol{\sigma}(Z)||_2^2 + \lambda\sum_i \sigma_i(Z)\right) \end{aligned} \tag{12}$$

Observe that the optimization problem in the right hand side of (12) is a separable vector optimization problem. We claim that the optimum solutions of the two problems appearing in (12) are in fact equal. To see this, let

$$\widehat{\boldsymbol{\sigma}(Z)} = \underset{\boldsymbol{\sigma}(Z):\|\boldsymbol{\sigma}(Z)\|_0\leq r}{\arg\min} \left(\tfrac{1}{2}||\boldsymbol{\sigma}(X)-\boldsymbol{\sigma}(Z)||_2^2 + \lambda\sum_i \sigma_i(Z)\right).$$

If the SVD of X is given by UDV^T, then the choice $\widehat{Z} = U\text{diag}(\widehat{\boldsymbol{\sigma}(Z)})V^T$ satisfies

$$\text{rank}(\widehat{Z}) \leq r \;\text{ and }\; F_\lambda(\widehat{Z}) = f_\lambda(\widehat{\boldsymbol{\sigma}(Z)})$$

This shows that:

$$\begin{aligned} &\min_{Z:\,\text{rank}(Z)\leq r} \left(\tfrac{1}{2}||X-Z||_F^2 + \lambda||Z||_*\right) \\ = &\min_{\boldsymbol{\sigma}(Z):\|\boldsymbol{\sigma}(Z)\|_0\leq r} \left(\tfrac{1}{2}||\boldsymbol{\sigma}(X)-\boldsymbol{\sigma}(Z)||_2^2 + \lambda\sum_i \sigma_i(Z)\right) \end{aligned} \tag{13}$$

and thus concludes the proof of the theorem.

■

This generalizes a similar result where there is no rank restriction, and the problem is convex in Z. For $r < \min(m,n)$, Problem (9) is not convex in Z, but the solution can be characterized in terms of the SVD of X.

The second theorem relates this problem to the corresponding matrix factorization problem

Theorem 2 *Let $X_{m\times n}$ be a matrix (fully observed), and let $0 < r \leq \min(m,n)$. Consider the optimization problem*

$$\min_{A_{m\times r},\, B_{n\times r}} \frac{1}{2}\left\|X - AB^T\right\|_F^2 + \frac{\lambda}{2}\left(\|A\|_F^2 + \|B\|_F^2\right) \tag{14}$$

A solution is given by $\widehat{A} = U_r\mathcal{S}_\lambda(D_r)^{\frac{1}{2}}$ and $\widehat{B} = V_r\mathcal{S}_\lambda(D_r)^{\frac{1}{2}}$, and all solutions satisfy $\widehat{A}\widehat{B}^T = \widehat{Z}$, where, $\widehat{Z}$ is as given in Problem (10).

We make use of the following lemma from Srebro et al. (2005); Mazumder et al. (2010), which we give without proof:

Lemma 1

$$\|Z\|_* = \min_{A,B:Z=AB^T} \frac{1}{2}\left(\|A\|_F^2 + \|B\|_F^2\right)$$

Proof (of Theorem 2). Using Lemma 1, we have that

$$\begin{aligned}&\min_{A_{m\times r},\, B_{n\times r}} \frac{1}{2}\left\|X - AB^T\right\|_F^2 + \frac{\lambda}{2}\|A\|_F^2 + \frac{\lambda}{2}\|B\|_F^2 \\ &= \min_{Z:\mathrm{rank}(Z)\le r} \frac{1}{2}\|X - Z\|_F^2 + \lambda\|Z\|_*\end{aligned}$$

The conclusions follow from Theorem 1. ■

Note, in both theorems the solution might have rank less than r.

Inspired by the alternating subspace iteration algorithm (a.k.a. Orthogonal Iterations, Chapter 8, Golub and Van Loan, 2012) for the reduced-rank SVD, we present Algorithm 2.1, an alternating ridge-regression algorithm for finding the solution to Problem (9).

Remarks

1. At each step the algorithm keeps the current solution in "SVD" form, by representing A and B in terms of orthogonal matrices. The computational effort needed to do this is exactly that required to perform each ridge regression, and once done makes the subsequent ridge regression trivial.

2. The proof of convergence of this algorithm is essentially the same as that for an alternating subspace algorithm, a.k.a. Orthogonal Iterations (Chapter 8, Thm 8.2.2; Golub and Van Loan, 2012) (without shrinkage).

3. In principle step (7) is not necessary, but in practice it cleans up the rank nicely.

4. This algorithm lends itself naturally to distributed computing for very large matrices X; X can be chunked into smaller blocks, and the left and right matrix multiplies can be chunked accordingly. See Section 8.

5. There are many ways to check for convergence. Suppose we have a pair of iterates (U, D^2, V) (old) and $(\tilde{U}, \tilde{D}^2, \tilde{V})$ (new), then the relative change in Frobenius norm is given by

$$\begin{aligned}\nabla F &= \frac{||UD^2V^T - \tilde{U}\tilde{D}^2\tilde{V}^T||_F^2}{||UD^2V^T||_F^2} \\ &= \frac{\mathrm{tr}(D^4) + \mathrm{tr}(\tilde{D}^4) - 2\,\mathrm{tr}(D^2U^T\tilde{U}\tilde{D}^2\tilde{V}^TV)}{\mathrm{tr}(D^4)},\end{aligned} \tag{19}$$

   which is not expensive to compute.

6. If X is sparse, then the left and right matrix multiplies can be achieved efficiently by using sparse matrix methods.

Algorithm 2.1 Rank-Restricted Soft SVD

1. Initialize $A = UD$ where $U_{m\times r}$ is a randomly chosen matrix with orthonormal columns and $D = I_r$, the $r \times r$ identity matrix.

2. Given A, solve for B:

$$\underset{B}{\text{minimize}}\, ||X - AB^T||_F^T + \lambda||B||_F^2. \tag{15}$$

   This is a multiresponse ridge regression, with solution

$$\tilde{B}^T = (D^2 + \lambda I)^{-1} DU^T X. \tag{16}$$

   This is simply matrix multiplication followed by coordinate-wise shrinkage.

3. Compute the SVD of $\tilde{B}D = \tilde{V}\tilde{D}^2R^T$, and let $V \leftarrow \tilde{V}$, $D \leftarrow \tilde{D}$, and $B = VD$.

4. Given B, solve for A:

$$\underset{A}{\text{minimize}}\, ||X - AB^T||_F^T + \lambda||A||_F^2. \tag{17}$$

   This is also a multiresponse ridge regression, with solution

$$\tilde{A} = XVD(D^2 + \lambda I)^{-1}. \tag{18}$$

   Again matrix multiplication followed by coordinate-wise shrinkage.

5. Compute the SVD of $\tilde{A}D = \tilde{U}\tilde{D}^2R^T$, and let $U \leftarrow \tilde{U}$, $D \leftarrow \tilde{D}$, and $A = UD$.

6. Repeat steps (2)–(5) until convergence of AB^T.

7. Compute $M = XV$, and then it's SVD: $M = UD_\sigma R^T$. Then output U, $V \leftarrow VR$ and $\mathcal{S}_\lambda(D_\sigma) = \text{diag}[(\sigma_1 - \lambda)_+, \ldots, (\sigma_r - \lambda)_+]$.

7. Likewise, if X is sparse, but has been column and/or row centered (see Section 9), it can be represented in "sparse plus low rank" form; once again left and right multiplication of such matrices can be done efficiently.

An interesting feature of this algorithm is that a reduced rank SVD of X is available from the solution, with the rank determined by the particular value of λ used. The singular values would have to be corrected by adding λ to each. There is empirical evidence that this is faster than without shrinkage, with accuracy biased more toward the larger singular values.

3. The `softImpute-ALS` Algorithm

Now we return to the case where X has missing values, and the non-missing entries are indexed by the set Ω. We present Algorithm 3.1 (`softImpute-ALS`) for solving Problem (6):

$$\underset{A,B}{\text{minimize}} \quad \|P_\Omega(X - AB^T)\|_F^2 + \lambda(\|A\|_F^2 + \|B\|_F^2).$$

where $A_{m\times r}$ and $B_{n\times r}$ are each of rank at most $r \le \min(m, n)$.

The algorithm exploits the decomposition

$$P_\Omega(X - AB^T) = P_\Omega(X) + P_\Omega^\perp(AB^T) - AB^T. \tag{24}$$

Suppose we have current estimates for A and B, and we wish to compute the new $\widetilde{B}$. We will replace the first occurrence of AB^T in the right-hand side of (24) with the current estimates, leading to a *filled in* $X^* = P_\Omega(X) + P_\Omega^\perp(AB^T)$, and then solve for $\widetilde{B}$ in

$$\underset{\widetilde{B}}{\text{minimize}} \quad \|X^* - A\widetilde{B}\|_F^2 + \lambda\|\widetilde{B}\|_F^2.$$

Using the same notation, we can write (importantly)

$$X^* = P_\Omega(X) + P_\Omega^\perp(AB^T) = \big(P_\Omega(X) - P_\Omega(AB^T)\big) + AB^T; \tag{25}$$

This is the efficient *sparse + low-rank* representation for high-dimensional problems; efficient to store and also efficient for left and right multiplication.

Remarks

1. This algorithm is a slight modification of Algorithm 2.1, where in step 2(a) we use the latest imputed matrix X^* rather than X.

2. The computations in step 2(b) are particularly efficient. In (22) we use the current version of A and B to predict at the observed entries Ω, and then perform a multiplication of a sparse matrix on the left by a skinny matrix, followed by rescaling of the rows. In (23) we simply rescale the rows of the previous version for B^T.

3. After each update, we maintain the integrity of the current solution. By Lemma 1 we know that the solution to

$$\underset{A,\, B\,:AB^T=\tilde{A}\tilde{B}^T}{\text{minimize}} (\|A\|_F^2 + \|B\|_F^2) \tag{26}$$

Algorithm 3.1 Rank-Restricted Efficient Maximum-Margin Matrix Factorization: `softImpute-ALS`

1. Initialize $A = UD$ where $U_{m\times r}$ is a randomly chosen matrix with orthonormal columns and $D = I_r$, the $r \times r$ identity matrix, and $B = VD$ with $V = 0$. Alternatively, any prior solution $A = UD$ and $B = VD$ could be used as a warm start.

2. Given $A = UD$ and $B = VD$, approximately solve

$$\underset{\widetilde{B}}{\text{minimize}}\, \frac{1}{2}\|P_\Omega(X - A\widetilde{B}^T)\|_F^T + \frac{\lambda}{2}\|\widetilde{B}\|_F^2 \tag{20}$$

   to update B. We achieve that with the following steps:

   (a) Let $X^* = \left(P_\Omega(X) - P_\Omega(AB^T)\right) + AB^T$, stored as *sparse plus low-rank*.

   (b) Solve

$$\underset{\widetilde{B}}{\text{minimize}}\, \frac{1}{2}\|X^* - A\widetilde{B}^T\|_F^2 + \frac{\lambda}{2}\|\widetilde{B}\|_F^2, \tag{21}$$

   with solution

$$\begin{aligned} \widetilde{B}^T &= (D^2 + \lambda I)^{-1} DU^T X^* \\ &= (D^2 + \lambda I)^{-1} DU^T \left(P_\Omega(X) - P_\Omega(AB^T)\right) \qquad (22) \\ &\quad + (D^2 + \lambda I)^{-1} D^2 B^T. \qquad (23) \end{aligned}$$

   (c) Use this solution for $\widetilde{B}$ and update V and D:

      i. Compute the SVD decomposition $\tilde{B}D = \tilde{U}\tilde{D}^2\tilde{V}^T$;

      ii. $V \leftarrow \tilde{U}$, and $D \leftarrow \tilde{D}$.

3. Given $B = VD$, solve for A. By symmetry, this is equivalent to step 2, with X^T replacing X, and B and A interchanged.

4. Repeat steps (2)–(3) until convergence.

5. Compute $M = X^*V$, and then it's SVD: $M = UD_\sigma R^T$. Then output U, $V \leftarrow VR$ and $D_{\sigma,\lambda} = \text{diag}[(\sigma_1 - \lambda)_+, \ldots, (\sigma_r - \lambda)_+]$.

is given by the SVD of $\tilde{A}\tilde{B}^T = UD^2V^T$, with $A = UD$ and $B = VD$. Our iterates maintain this each time A or B changes in step 2(c), with no additional significant computational cost.

4. The final step is as in Algorithm 2.1. We know the solution should have the form of a soft-thresholded SVD. The alternating ridge regression might not exactly reveal the rank of the solution. This final step tends to clean this up, by revealing exact zeros after the soft-thresholding.

5. In Section 5 we discuss (the lack of) optimality guarantees of fixed points of Algorithm 3.1 (in terms of criterion (1)). We note that the output of `softImpute-ALS` can easily be piped into `softImpute` as a warm start. This typically exits almost immediately in our `R` package `softImpute`.

4. Broader Perspective and Related Work

Block coordinate descent (for example, Bertsekas, 1999) is a classical method in optimization that is widely used in the statistical and machine learning communities (Hastie et al., 2009). This is useful especially when the optimization problems associated with each block is relatively simple. The algorithm presented in this paper is a stylized variant of block coordinate descent. At a high level *vanilla* block coordinate descent (which we call `ALS`) applied to Problem (6) performs a complete minimization wrt one variable with the other fixed, before it switches to over the other variable. `softImpute-ALS` instead, does a *partial* minimization of a very specific form. Razaviyayn et al. (2013) study convergence properties of generalized block-coordinate methods that apply to a fairly large class of problems. The same paper presents asymptotic convergence guarantees, i.e., the iterates converge to a stationary point (Bertsekas, 1999). Asymptotic convergence is fairly straightforward to establish for `softimpute-ALS`. We also describe global convergence rate guarantees[2] for `softimpute-ALS` in terms of various metrics of convergence to a stationary point. Perhaps more interestingly, we connect the properties of the stationary points of the non-convex Problem (6) to the minimizers of the convex Problem (1), which seems to be well beyond the scope and intent of Razaviyayn et al. (2013).

Variations of alternating-minimization strategies are popularly used in the context of matrix completion (Chen et al., 2012; Koren et al., 2009; Zhou et al., 2008). Jain et al. (2013) analyze the statistical properties of vanilla alternating-minimization algorithms for Problem (6) with $\lambda = 0$, i.e.,

$$\underset{A,B}{\text{minimize}} \quad \|P_\Omega(X - AB^T)\|_F^2,$$

where, one attempts to minimize the above function via alternating least squares *i.e.* first minimizing with respect to A (with B fixed) and vice-versa. They establish statistical performance guarantees of the alternating strategy under incoherence conditions on the singular vectors of the *underlying* low-rank matrix—the assumptions are similar in spirit

2. By global convergence rate, we mean an upper bound on the maximum number of iterations that need to be taken by an algorithm to reach an ϵ-accurate first-order stationary point. This rate applies for *any* starting point of the algorithm.

to the work of Candès and Tao (2009); Candès and Recht (2008). However, as pointed out by Jain et al. (2013), their alternating-minimization methods typically require $|\Omega|$ to be larger than than required by convex optimization based methods (Candès and Recht, 2008). We refer the interested reader to more recent work of Hardt (2014), analyzing the statistical properties of alternating minimization methods.

The flavor of our present work is in spirit different from that described above (Jain et al., 2013; Hardt, 2014). Our main goal here is to develop non-convex algorithms for the optimization of Problem (6) for arbitrary λ and rank r. A special case of our framework corresponds to the case where Problem (6) is equivalent to Problem (1), for proper choices of r, λ. In this particular case, we study in Section 5 when our algorithm `softImpute-ALS` converges to a global minimizer of Problem (1)—this can be verified by a minor check that requires computing the low-rank SVD of a matrix that can be written as the sum of a sparse and low-rank matrix. Thus `softImpute-ALS` can be thought of a *non-convex* algorithm that *solves* the convex nuclear norm regularized Problem (1). Hence `softImpute-ALS` inherits statistical properties of the convex Problem (1) as established in Candès and Tao (2009); Candès and Recht (2008). We have also demonstrated in Figures 1 and 3 that `softimpute-ALS` is much faster than the alternating least squares schemes analyzed in Jain et al. (2013); Hardt (2014).

Note that the use of non-convex methods to obtain minimizers of convex problems have been studied in Burer and Monteiro (2005); Journée et al. (2010). The authors study non-linear optimization algorithms using non-convex matrix factorization formulations to obtain global minimizers of convex SDPs. The results presented in the aforementioned papers also requires one to check whether a stationary point is a *local minimizer*—this typically requires checking the positive definiteness of a matrix of size $O(mr + nr) \times O(mr + nr)$ and can be computationally demanding if the problem size is large. In contrast, the condition (derived in this paper) that needs to be checked to certify whether `softimpute-ALS`, upon convergence, has reached the global solution to the convex optimization Problem (1), is fairly intuitive and straightforward.

5. Algorithmic Convergence Analysis

In this section we investigate the theoretical properties of the `softImpute-ALS` algorithm in the context of Problems (1) and (6).

We show that the `softImpute-ALS` algorithm converges to a first order stationary point for Problem (6) at a rate of $O(1/K)$, where K denotes the number of iterations of the algorithm. We also discuss the role played by λ in the convergence rates. We establish the limiting properties of the estimates produced by the `softImpute-ALS` algorithm: properties of the limit points of the sequence (A_k, B_k) in terms of Problems (1) and (6). We show that for any r in Problem (6) the sequence produced by the `softImpute-ALS` algorithm leads to a decreasing sequence of objective values for the convex Problem (1). A fixed point of the `softImpute-ALS` problem need not correspond to the minimum of the convex Problem (1). We derive simple necessary and sufficient conditions that must be satisfied for a stationary point of the algorithm to be a minimum for the Problem (1)—the conditions can be verified by a simple structured low-rank SVD computation.

We begin the section with a formal description of the updates produced by the algorithm in terms of the original objective function (6) and its majorizers (27) and (28). Theorem 3 establishes that the updates lead to a decreasing sequence of objective values $F(A_k, B_k)$ in (6). Section 5.1 (Theorem 4 and Corollary 1) derives the finite-time convergence rate properties of the proposed algorithm `softImpute-ALS`. Section 5.2 provides descriptions of the first order stationary conditions for Problem (6), the fixed points of the algorithm `softImpute-ALS` and the limiting behavior of the sequence $(A_k, B_k), k \geq 1$ as $k \to \infty$. Section 5.3 (Lemma 4) investigates the implications of the updates produced by `softImpute-ALS` for Problem (6) in terms of the Problem (1). Section 5.3.1 (Theorem 6) studies the stationarity conditions for Problem (6) vis-a-vis the optimality conditions for the convex Problem (1).

The `softImpute-ALS` algorithm may be thought of as an EM or more generally a MM-style algorithm (majorization minimization), where every imputation step leads to an upper bound to the training error part of the loss function. The resultant loss function is minimized wrt A—this leads to a partial minimization of the objective function wrt A. The process is repeated with the other factor B, and continued till convergence.

Recall the objective function in Problem (6):

$$F(A, B) := \frac{1}{2} \left\| P_\Omega(X - AB^T) \right\|_F^2 + \frac{\lambda}{2} \|A\|_F^2 + \frac{\lambda}{2} \|B\|_F^2 .$$

We define the surrogate functions

$$\begin{aligned} Q_A(Z_1|A, B) &:= \frac{1}{2} \left\| P_\Omega(X - Z_1B^T) + P_\Omega^\perp(AB^T - Z_1B^T) \right\|_F^2 \\ &\quad + \frac{\lambda}{2} \|Z_1\|_F^2 + \frac{\lambda}{2} \|B\|_F^2 \qquad (27) \\ Q_B(Z_2|A, B) &:= \frac{1}{2} \left\| P_\Omega(X - AZ_2^T) + P_\Omega^\perp(AB^T - AZ_2^T) \right\|_F^2 \\ &\quad + \frac{\lambda}{2} \|A\|_F^2 + \frac{\lambda}{2} \|Z_2\|_F^2 . \qquad (28) \end{aligned}$$

Consider the function $g(AB^T) := \frac{1}{2} \left\| P_\Omega(X - AB^T) \right\|_F^2$ which is the training error as a function of the outer-product $Z = AB^T$, and observe that for any $Z, \overline{Z}$ we have:

$$\begin{aligned} g(Z) \leq & \frac{1}{2} \left\| P_\Omega(X - Z) + P_\Omega^\perp(\overline{Z} - Z) \right\|_F^2 \\ = & \frac{1}{2} \left\| \left(P_\Omega(X) + P_\Omega^\perp(\overline{Z}) \right) - Z \right\|_F^2 \end{aligned} \qquad (29)$$

where, equality holds above at $Z = \overline{Z}$. This leads to the following simple but important observations:

$$Q_A(Z_1|A, B) \geq F(Z_1, B), \qquad Q_B(Z_2|A, B) \geq F(A, Z_2), \qquad (30)$$

suggesting that $Q_A(Z_1|A, B)$ is a majorizer of $F(Z_1, B)$ (as a function of Z_1); similarly, $Q_B(Z_2|A, B)$ majorizes $F(A, Z_2)$. In addition, equality holds as follows:

$$Q_A(A|A, B) = F(A, B) = Q_B(B|A, B). \qquad (31)$$

We also define $X^*_{A,B} = P_\Omega(X) + P^\perp_\Omega(AB^T)$. Using these definitions, we can succinctly describe the `softImpute-ALS` algorithm in Algorithm 5.1. This is almost equivalent to Algorithm 3.1, but more convenient for theoretical analysis. It has the orthogonalization and redistribution of $\tilde{D}$ in step 3 removed, and step 5 removed. Observe that the

Algorithm 5.1 `softImpute-ALS`

Inputs: Data matrix X, initial iterates A_0 and B_0, and $k = 0$.
Outputs: (A^*, B^*) an estimate of the minimizer of Problem (6)

Repeat until Convergence

1. $k \leftarrow k + 1$.
2. $X^* \leftarrow P_\Omega(X) + P^\perp_\Omega(AB^T) = P_\Omega(X - AB^T) + AB^T$
3. $A \leftarrow X^*B(B^TB + \lambda I)^{-1} = \arg\min_{Z_1} \; Q_A(Z_1|A, B)$.
4. $X^* \leftarrow P_\Omega(X) + P^\perp_\Omega(AB^T)$
5. $B \leftarrow X^{*T}A(A^TA + \lambda I)^{-1} = \arg\min_{Z_2} \; Q_B(Z_2|A, B)$.

`softImpute-ALS` algorithm can be described as the following iterative procedure:

$$A_{k+1} \in \arg\min_{Z_1} \; Q_A(Z_1|A_k, B_k) \tag{32}$$

$$B_{k+1} \in \arg\min_{Z_2} \; Q_B(Z_2|A_{k+1}, B_k). \tag{33}$$

We will use the above notation in our proof.

We can easily establish that `softImpute-ALS` is a descent method, or its iterates never increase the function value.

Theorem 3 *Let $\{(A_k, B_k)\}$ be the iterates generated by* `softImpute-ALS`*. The function values are monotonically decreasing,*

$$F(A_k, B_k) \geq F(A_{k+1}, B_k) \geq F(A_{k+1}, B_{k+1}), \;\; k \geq 1.$$

Proof Let the current iterate estimates be (A_k, B_k). We will first consider the update in A, leading to A_{k+1}, as defined in (32).

$$\min_{Z_1} \; Q_A(Z_1|A_k, B_k) \leq Q_A(A_k|A_k, B_k) = F(A_k, B_k)$$

Note that, $\min_{Z_1} \; Q_A(Z_1|A_k, B_k) = Q_A(A_{k+1}|A_k, B_k)$, by definition of A_{k+1} in (32).

Using (30) we get that $Q_A(A_{k+1}|A_k, B_k) \geq F(A_{k+1}, B_k)$. Putting together the pieces we get: $F(A_k, B_k) \geq F(A_{k+1}, B_k)$.

Using an argument exactly similar to the above for the update in B we have:

$$F(A_{k+1}, B_k) = Q_B(B_k|A_{k+1}, B_k) \geq Q_B(B_{k+1}|A_{k+1}, B_k) \geq F(A_{k+1}, B_{k+1}). \tag{34}$$

This establishes that $F(A_k, B_k) \geq F(A_{k+1}, B_{k+1})$ for all k, thereby completing the proof of the theorem. ■

5.1 `softImpute-ALS`: Rates of Convergence

The previous section derives some elementary properties of the `softImpute-ALS` algorithm, namely the updates lead to a decreasing sequence of objective values. We will now derive some results that inform us about the rate at which the `softImpute-ALS` algorithm reaches a stationary point.

We begin with the following lemma, which presents a lower bound on the successive difference in objective values of $F(A, B)$,

Lemma 2 *Let (A_k, B_k) denote the values of the factors at iteration k. We have the following:*

$$\begin{aligned} F(A_k, B_k) - F(A_{k+1}, B_{k+1}) \geq & \tfrac{1}{2}\left(\|(A_k - A_{k+1})B_k^T\|_F^2 + \|A_{k+1}(B_{k+1} - B_k)^T\|_F^2\right) \\ & + \frac{\lambda}{2}\left(\|A_k - A_{k+1}\|_F^2 + \|B_{k+1} - B_k\|_F^2\right) \end{aligned} \tag{35}$$

Proof *See Section A.1 for the proof.* ■

For any two matrices A and B respectively define A^+, B^+ as follows:

$$A^+ \in \arg\min_{Z_1} \; Q_A(Z_1|A, B), \qquad B^+ \in \arg\min_{Z_2} \; Q_B(Z_2|A, B) \tag{36}$$

We will consequently define the following:

$$\begin{aligned} \Delta\left((A, B), \left(A^+, B^+\right)\right) := & \tfrac{1}{2}\left(\|(A - A^+)B^T\|_F^2 + \|A^+(B - B^+)^T\|_F^2\right) \\ & + \frac{\lambda}{2}\left(\|A - A^+\|_F^2 + \|B - B^+\|_F^2\right) \end{aligned} \tag{37}$$

Lemma 3 *$\Delta\left((A, B), (A^+, B^+)\right) = 0$ iff A, B is a fixed point of* `softImpute-ALS`.
Proof *See Section A.2, for a proof.* ■

We will use the following notation

$$\eta_k := \Delta\left((A_k, B_k), (A_{k+1}, B_{k+1})\right) \tag{38}$$

Thus η_k can be used to quantify how close (A_k, B_k) is from a stationary point.

If $\eta_k > 0$ it means that Algorithm `softImpute-ALS` will make progress in improving the quality of the solution. As a consequence of the monotone decreasing property of the sequence of objective values $F(A_k, B_k)$ and Lemma 2, we have that, $\eta_k \to 0$ as $k \to \infty$. The following theorem characterizes the rate at which η_k converges to zero.

Theorem 4 *Let $(A_k, B_k), k \geq 1$ be the sequence generated by the* `softImpute-ALS` *algorithm. The decreasing sequence of objective values $F(A_k, B_k)$ converges to $F^\infty \geq 0$ (say) and the quantities $\eta_k \to 0$.*

Furthermore, we have the following finite convergence rate of the `softImpute-ALS` *algorithm:*

$$\min_{1\leq k\leq K} \eta_k \leq \left(F(A_1, B_1) - F^\infty)\right)/K \tag{39}$$

Proof *See Section A.3* ■

The above theorem establishes a $O(\frac{1}{K})$ convergence rate of `softImpute-ALS`; in other words, for any $\epsilon > 0$, we need at most $K = O(\frac{1}{\epsilon})$ iterations to arrive at a point (A_{k^*}, B_{k^*}) such that $\eta_{k^*} \leq \epsilon$, where, $1 \leq k^* \leq K$.

Note that Theorem 4 establishes convergence of the algorithm for *any* value of $\lambda \geq 0$. We found in our numerical experiments that the value of λ has an important role to play in the speed of convergence of the algorithm. In the following corollary, we provide convergence rates that make the role of λ explicit.

The following corollary employs three different distance measures to measure the closeness of a point from stationarity.

Corollary 1 *Let $(A_k, B_k), k \geq 1$ be defined as above. Assume that for all $k \geq 1$*

$$\ell^U \mathbf{I} \succeq B_k^T B_k \succeq \ell^L \mathbf{I}, \;\; \ell^U \mathbf{I} \succeq A_k^T A_k \succeq \ell^L \mathbf{I}, \tag{40}$$

where, ℓ^U, ℓ^L are constants independent of k.

Then we have the following:

$$\min_{1\leq k\leq K} \left(\|(A_k - A_{k+1})\|_F^2 + \|B_k - B_{k+1}\|_F^2\right) \leq \frac{2}{(\ell^L + \lambda)} \left(\frac{F(A_1, B_1) - F^\infty}{K}\right) \tag{41}$$

$$\min_{1\leq k\leq K} \left(\|(A_k - A_{k+1})B_k^T\|_F^2 + \|A_{k+1}(B_k - B_{k+1})^T\|_F^2\right) \leq \frac{2\ell^U}{\lambda + \ell_U} \left(\frac{F(A_1, B_1) - F^\infty}{K}\right) \tag{42}$$

$$\min_{1\leq k\leq K} \left(\|\nabla_A f(A_k, B_k)\|^2 + \|\nabla_B f(A_{k+1}, B_k)\|^2\right) \leq \frac{2(\ell^U)^2}{(\ell^L + \lambda)} \left(\frac{F(A_1, B_1) - F^\infty}{K}\right) \tag{43}$$

where, $\nabla_A f(A, B)$ (respectively, $\nabla_B f(A, B)$) denotes the partial derivative of $F(A, B)$ wrt A (respectively, B).

Proof *See Section A.4.* ■

Inequalities (41)–(43) are statements about different notions of distances between successive iterates. These may be employed to understand the convergence rate of `softImpute-ALS`.

Assumption (40) is a minor one. While it may not be straightforward to estimate ℓ^U prior to running the algorithm, a finite value of ℓ^U is guaranteed as soon as $\lambda > 0$. The lower bound $\ell_L > 0$, if both $A_1 \in \Re^{m\times r}, B_1 \in \Re^{n\times r}$ have rank r and the rank remains

the same across the iterates. If the solution to Problem (6) has a rank smaller than r, then $\ell_L = 0$, however, this situation is typically avoided since a small value of r leads to lower computational cost per iteration of the `softImpute-ALS` algorithm. The constants appearing as a part of the rates in (41)–(43) are dependent upon λ. The constants are smaller for larger values of λ. Finally we note that the algorithm does not require any information about the constants ℓ^L, ℓ^U appearing as a part of the rate estimates.

5.2 `softImpute-ALS`: Asymptotic Convergence

In this section we derive some properties of the limiting behavior of the sequence (A_k, B_k), in particular we examine some elementary properties of the limit points of the sequence (A_k, B_k).

At the beginning, we recall the notion of first order stationarity of a point A_*, B_*. We say that A_*, B_* is said to be a first order stationary point for the Problem (6) if the following holds:

$$\nabla_A f(A_*, B_*) = 0, \quad \nabla_B f(A_*, B_*) = 0. \tag{44}$$

An equivalent restatement of condition (44) is:

$$A_* \in \arg\min_{Z_1} \; Q_A(Z_1|A_*, B_*), \quad B_* \in \arg\min_{Z_2} \; Q_B(Z_2|A_*, B_*), \tag{45}$$

i.e., A_*, B_* is a fixed point of the `softImpute-ALS` algorithm updates.

We now consider uniqueness properties of the limit points of $(A_k, B_k), k \geq 1$. Even in the fully observed case, the stationary points of Problem (6) are not unique in A_*, B_*; due to orthogonal invariance. Addressing convergence of (A_k, B_k) becomes trickier if two singular values of $A_* B_*^T$ are tied. In this vein we have the following result:

Theorem 5 *Let $\{(A_k, B_k)\}_k$ be the sequence of iterates generated by Algorithm 5.1. For $\lambda > 0$, we have:*

(a) Every limit point of $\{(A_k, B_k)\}_k$ is a stationary point of Problem (6).

(b) Let B_* be any limit point of the sequence $B_k, k \geq 1$, with $B_\nu \to B_*$, where, ν is a subsequence of $\{1, 2, \ldots, \}$. Then the sequence A_ν converges.

Similarly, let A_* be any limit point of the sequence $A_k, k \geq 1$, with $A_\mu \to B_*$, where, μ is a subsequence of $\{1, 2, \ldots, \}$. Then the sequence B_μ converges.

Proof *See Section A.5* ■

The above theorem is a partial result about the uniqueness of the limit points of the sequence A_k, B_k. The theorem implies that if the sequence A_k converges, then the sequence B_k must converge and vice-versa. More generally, for every limit point of A_k, the associated B_k (sub)sequence will converge. The same result holds true for the sequence B_k.

Remark 1 *Note that the condition $\lambda > 0$ is enforced due to technical reasons so that the sequence (A_k, B_k) remains bounded. If $\lambda = 0$, then $A \leftarrow cA$ and $B \leftarrow \frac{1}{c}B$ for any $c > 0$, leaves the objective function unchanged. Thus one may take $c \to \infty$ making the sequence of updates unbounded without making any change to the values of the objective function.*

5.3 Implications of `softImpute-ALS` updates in terms of Problem (1)

The sequence (A_k, B_k) generated by Algorithm (5.1) are geared towards minimizing criterion (6), it is interesting to explore what implications the sequence might have for the convex Problem (1). In particular, we know that $F(A_k, B_k)$ is decreasing—does this imply a monotone sequence $H(A_k B_k^T)$? We show below that it is indeed possible to obtain a monotone decreasing sequence $H(A_k B_k^T)$ with a minor modification. These modifications are exactly those implemented in Algorithm 3.1 in step 3.

The idea that plays a crucial role in this modification is the following inequality (for a proof see Mazumder et al. (2010); see also remark 3 in Section 3):

$$\|AB^T\|_* \leq \tfrac{1}{2}(\|A\|_F^2 + \|B\|_F^2).$$

Note that equality holds above if we take a particular choice of A and B given by:

$$A = UD^{1/2}, B = VD^{1/2}, \quad \text{where,} \quad AB^T = UDV^T \quad \text{(SVD)}, \tag{46}$$

is the SVD of AB^T. The above observation implies that if (A_k, B_k) is generated by `softImpute-ALS` then

$$F(A_k, B_k) \geq H(A_k B_k^T)$$

with equality holding if A_k, B_k^T are represented via (46). Note that this re-parametrization does not change the training error portion of the objective $F(A_k, B_k)$, but decreases the ridge regularization term—and hence decreases the overall objective value when compared to that achieved by `softImpute-ALS` without the reparametrization (46).

We thus have the following Lemma:

Lemma 4 *Let the sequence (A_k, B_k) generated by `softImpute-ALS` be stored in terms of the factored SVD representation (46). This results in a decreasing sequence of objective values in the nuclear norm penalized Problem (1):*

$$H(A_k B_k^T) \geq H(A_{k+1} B_{k+1}^T)$$

with $H(A_k B_k^T) = F(A_k, B_k)$, for all k. The sequence $H(A_k B_k^T)$ thus converges to F^∞.

Note that, F^∞ need not be the minimum of the convex Problem (1). It is easy to see this, by taking r to be smaller than the rank of the optimal solution to Problem (1).

5.3.1 A Closer Look at the Stationary Conditions

In this section we inspect the first order stationary conditions of the non-convex Problem (6) alongside those for the convex Problem (1). We will see that a first order stationary point of the convex Problem (1) leads to factors (A, B) that are stationary for Problem (6). However, the converse of this statement need not be true in general. However, given an estimate delivered by `softImpute-ALS` (upon convergence) it is easy to verify whether it is a solution to Problem (1).

Note that Z^* is the optimal solution to the convex Problem (1) iff:

$$\partial H(Z^*) = P_\Omega(Z^* - X) + \lambda \operatorname{sgn}(Z^*) = 0,$$

where, $\text{sgn}(Z^*)$ is a sub-gradient of the nuclear norm $\|Z\|_*$ at Z^*. Using the standard characterization (Lewis, 1996) of $\text{sgn}(Z^*)$ the above condition is equivalent to:

$$P_\Omega(Z^* - X) + \lambda U_* \,\text{sgn}(D^*) V_*^T = 0 \tag{47}$$

where, the full SVD of Z^* is given by $U_* D_* V_*^T$; $\text{sgn}(D^*)$ is a diagonal matrix with ith diagonal entry given by $\text{sgn}(d_{ii}^*)$, where, d_{ii}^* is the ith diagonal entry of D^*.

If a limit point $A_* B_*^T$ of the `softImpute-ALS` algorithm satisfies the stationarity condition (47) above, then it is the optimal solution of the convex problem. We note that $A_* B_*^T$ need not necessarily satisfy the stationarity condition (47).

(A, B) satisfy the stationarity conditions of `softImpute-ALS` if the following conditions are satisfied:

$$P_\Omega(AB^T - X)B + \lambda A = 0, \quad A^T(P_\Omega(AB^T - X)) + \lambda B^T = 0,$$

where, we assume that A, B are represented in terms of (46). This gives us:

$$P_\Omega(AB^T - X)V + \lambda U = 0, \quad U^T(P_\Omega(AB^T - X)) + \lambda V^T = 0, \tag{48}$$

where $AB^T = UDV^T$, being the reduced rank SVD i.e. all diagonal entries of D are strictly positive.

A stationary point of the convex problem corresponds to a stationary point of `softImpute-ALS`, as seen by a direct verification of the conditions above. In the following we investigate the converse: when does a stationary point of `softImpute-ALS` correspond to a stationary point of Problem (1); i.e. condition (47)? Towards this end, we make use of the ridged least-squares update used by `softImpute-ALS`. Assume that all matrices A_k, B_k have r rows.

At stationarity i.e. at a fixed point of `softImpute-ALS` we have the following:

$$A_* \in \quad \arg\min_A \ \tfrac{1}{2}\|P_\Omega(X - AB_*^T)\|_F^2 + \frac{\lambda}{2}\left(\|A\|_F^2 + \|B_*\|_F^2\right) \tag{49}$$

$$= \quad \arg\min_A \ \tfrac{1}{2}\| \left(P_\Omega(X) + P_\Omega^\perp(A_* B_*^T)\right) - AB_*^T\|_F^2 + \frac{\lambda}{2}\left(\|A\|_F^2 + \|B_*\|_F^2\right) \tag{50}$$

$$B_* \in \quad \arg\min_B \ \tfrac{1}{2}\|P_\Omega(X - A_* B^T)\|_F^2 + \frac{\lambda}{2}\left(\|A_*\|_F^2 + \|B\|_F^2\right) \tag{51}$$

$$= \quad \arg\min_B \ \tfrac{1}{2}\| \left(P_\Omega(X) + P_\Omega^\perp(A_* B_*^T)\right) - A_* B^T\|_F^2 + \frac{\lambda}{2}\left(\|A_*\|_F^2 + \|B\|_F^2\right) \tag{52}$$

Line (50) and (52) can be thought of doing alternating multiple ridge regressions for the fully observed matrix $P_\Omega(X) + P_\Omega^\perp(A_* B_*^T)$.

The above fixed point updates are very closely related to the following optimization problem:

$$\underset{A_{m\times r}, B_{m\times r}}{\text{minimize}} \ \tfrac{1}{2}\| \left(P_\Omega(X) + P_\Omega^\perp(A_* B_*^T)\right) - AB\|_F^2 + \frac{\lambda}{2}\left(\|A\|_F^2 + \|B\|_F^2\right) \tag{53}$$

The solution to (53) by Theorem 1 is given by the nuclear norm thresholding operation (with a rank r constraint) on the matrix $P_\Omega(X) + P_\Omega^\perp(A_* B_*^T)$:

$$\underset{Z:\text{rank}(Z)\le r}{\text{minimize}} \ \tfrac{1}{2}\| \left(P_\Omega(X) + P_\Omega^\perp(A_* B_*^T)\right) - Z\|_F^2 + \frac{\lambda}{2}\|Z\|_*. \tag{54}$$

Suppose the convex optimization Problem (1) has a solution Z^* with $\text{rank}(Z^*) = r^*$. Then, for $A_*B_*^T$ to be a solution to the convex problem the following conditions are sufficient:

(a) $r^* \leq r$

(b) A_*, B_* must be the global minimum of Problem (53). Equivalently, the outer product $A_*B_*^T$ must be the solution to the *fully observed* nuclear norm regularized problem:

$$A_*B_*^T \in \arg\min_Z \ \tfrac{1}{2}\| \left(P_\Omega(X) + P_\Omega^\perp(A_*B_*^T)\right) - Z\|_F^2 + \lambda\|Z\|_* \,. \tag{55}$$

The above condition (55) can be verified fairly easily; and requires doing a low-rank SVD of the matrix $P_\Omega(X) + P_\Omega^\perp(A_*B_*^T)$ as a direct application of Algorithm 2.1. This task is computationally attractive due to the "sparse plus low-rank structure" of the matrix: $P_\Omega(X) + P_\Omega^\perp(A_*B_*^T) = P_\Omega(X - A_*B_*^T) + A_*B_*^T$. We summarize the above discussion in the form of the following theorem, where we assume of course that $\lambda > 0$.

Theorem 6 *Let $A_k \in \Re^{m\times r}, B_k \in \Re^{n\times r}$ be the sequence generated by* `softImpute-ALS` *and let (A_*, B_*) denote a limit point of the sequence. Suppose that Problem* (1) *has a minimizer with rank at most r. If $Z_* = A_*B_*^T$ solves the* fully observed *nuclear norm regularized problem* (55)*, then Z_* is a solution to the convex Problem* (1)*.*

5.4 Computational Complexity and Comparison to `ALS`

The computational cost of `softImpute-ALS` can be broken down into three steps. First consider only the cost of the update to A. The first step is forming the matrix $X^* = P_\Omega(X - AB^T) + AB^T$, which requires $O(r|\Omega|)$ flops for the $P_\Omega(AB^T)$ part, while the second part is never explicitly formed. The matrix $B(B^TB + \lambda I)^{-1}$ requires $O(2nr^2 + r^3)$ flops to form; although we keep it in SVD factored form, the cost is the same. The multiplication $X^*B(B^TB + \lambda I)^{-1}$ requires $O(r|\Omega| + mr^2 + nr^2)$ flops, using the sparse plus low-rank structure of X^*. The total cost of an iteration is $O(2r|\Omega| + mr^2 + 3nr^2 + r^3)$.

As mentioned in Section 1, alternating least squares (`ALS`) is a popular algorithm for solving the matrix factorization problem in Equation (6); see Algorithm 5.2. The `ALS` algorithm is an instance of block coordinate descent applied to (6).

Recall that the updates for `ALS` are given by

$$A_{k+1} \in \arg\min_A \ F(A, B_k) \tag{56}$$

$$B_{k+1} \in \arg\min_B \ F(A_k, B), \tag{57}$$

and each row of A and B can be computed via a separate ridge regression. The cost for each ridge regression is $O(|\Omega_j|r^2 + r^3)$, so the cost of one iteration is $O(2|\Omega|r^2 + mr^3 + nr^3)$. Hence the cost of one iteration of `ALS` is r times more flops than one iteration of `softImpute-ALS`. We will see in the next sections that while `ALS` may decrease the criterion at each iteration more than `softImpute-ALS`, it tends to be slower because the cost is higher by a factor $O(r)$.

Algorithm 5.2 Alternating least squares `ALS`

Inputs: Data matrix X, initial iterates A_0 and B_0, and $k = 0$.
Outputs: $(A^*, B^*) = \arg\min_{A,B}\ F(A, B)$
Repeat until Convergence
  for i=1 to m **do**
    $A_i \leftarrow \left(\sum_{j\in\Omega_i} B_j B_j^T\right)^{-1}\left(\sum_{j\in\Omega_i} X_{ij}B_j\right)$
  end for
  for j=1 to n **do**
    $B_j \leftarrow \left(\sum_{i\in\Omega_j} A_i A_i^T\right)^{-1}\left(\sum_{i\in\Omega_j} X_{ij}A_i\right)$
  end for

Dependence of Computational Complexity on Ω: The computational guarantees derived in Section 5.1 present a worst-case viewpoint of the rate at which `softimpute-ALS` converge to an approximate stationary point—the results apply to *any* data and an arbitrary Ω. Tighter rates can be derived under additional assumptions. For example, for the special case where Ω corresponds to a fully observed matrix, `softimpute-ALS` becomes Algorithm 2.1. For $\lambda = 0$, Algorithm 2.1 with Ω fully observed becomes *exactly* equivalent to the Orthogonal Iteration algorithm of Golub and Van Loan (2012). Theorem 8.2.2 in Golub and Van Loan (2012) shows that the left orthogonal subspace corresponding to A converges to the left singular subspace of X, under the assumption that $\sigma_r(X) > \sigma_{r+1}(X)$—the rate is linear[3] and depends upon the ratio $\frac{\sigma_{r+1}(X)}{\sigma_r(X)}$. Similar results hold true for the left orthogonal subspace of B. Since the left subspaces of A and B generated by Algorithm 2.1 with $\lambda > 0$ are the same for $\lambda = 0$, the same linear rate of convergence holds true for Algorithm 2.1 for Problem (14).

For a general Ω it is not clear to us if the rates in Section 5.1 can be improved. However, for a sparse Ω the computational cost of every iteration of `softimpute-ALS` is significantly smaller than a dense observation pattern—the practical significance being that a large number of iterations can be performed at a very low cost.

6. Experiments

In this section we run some timing experiments on simulated and real datasets, and show performance results on the Netflix and MovieLens data.

6.1 Timing experiments

Figure 1 shows timing results on four datasets. The first three are simulation datasets of increasing size, and the last is the publicly available `MovieLens 100K` data. These experiments were all run in `R` using the `softImpute` package; see Section 7. Three methods are compared:

1. `ALS`— Alternating Least Squares as in Algorithm 5.2;

3. Convergence is measured in terms of the usual notion of distance between subspaces (Golub and Van Loan, 2012); and it is also assumed that the initialization is not completely orthogonal to the target subspace, which is typically met in practice due to the presence of round-off errors.

2. `softImpute-ALS` — our new approach, as defined in Algorithm 3.1 or 5.1;

3. `softImpute` — the original algorithm of Mazumder et al. (2010), as layed out in steps (2)–(4).

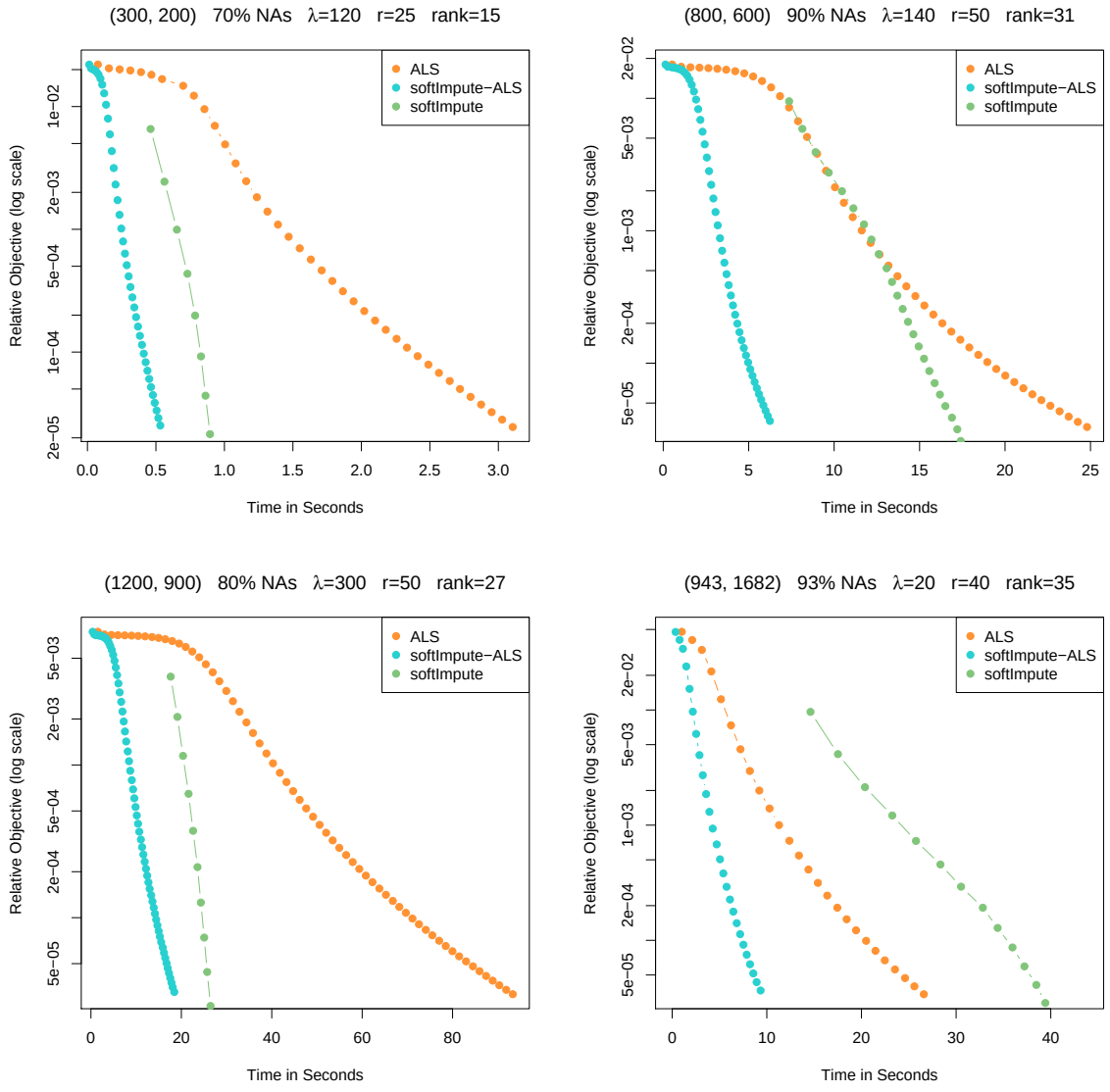


Figure 1: Four timing experiments. Each figure is labelled according to size ($m \times n$), percentage of missing entries (NAs), value of λ used, rank r used in the `ALS` iterations, and rank of solution found. The first three are simulation examples, with increasing dimension. The last is the `movielens 100K` data. In all cases, `softImpute-ALS` (blue) wins handily against `ALS` (orange) and `softImpute` (green).

We used an `R` implementation for each of these in order to make the fairest comparisons. In particular, algorithm `softImpute` requires a low-rank SVD of a complete matrix at each iteration. For this we used the function `svd.als` from our package, which uses alternating subspace iterations, rather than using other optimized code that is available for this task. Likewise, there exists optimized code for regular `ALS` for matrix completion, but instead we used our `R` version to make the comparisons fairer. We are trying to determine how the computational trade-offs play off, and thus need a level playing field.

Each subplot in Figure 6.1 is labeled according to the size of the problem, the fraction missing, the value of λ used, the operating rank of the algorithms r, and the rank of the solution obtained. All three methods involve alternating subspace methods; the first two are alternating ridge regressions, and the third alternating orthogonal regressions. These are conducted at the operating rank r, anticipating a solution of smaller rank. Upon convergence, `softImpute-ALS` performs step (5) in Algorithm 3.1, which can truncate the rank of the solution. Our implementation of `ALS` does the same.

For the three simulation examples, the data are generated from an underlying Gaussian factor model, with true ranks 50, 100, 100; the missing entries are then chosen at random. Their sizes are $(300, 200)$, $(800, 600)$ and $(1200, 900)$ respectively, with between 70–90% missing. The `MovieLens 100K` data has 100K ratings (1–5) for 943 users and 1682 movies, and hence is 93% missing.

We picked a value of λ for each of these examples (through trial and error) so that the final solution had rank less than the operating rank. Under these circumstances, the solution to the criterion (6) coincides with the solution to (1), which is unique under non-degenerate situations.

There is a fairly consistent message from each of these experiments. `softImpute-ALS` wins handily in each case, and the reasons are clear:

- Even though it uses more iterations than `ALS`, they are much cheaper to execute (by a factor $O(r)$).
- `softImpute` wastes time on its early SVD, even though it is far from the solution. Thereafter it uses warm starts for its SVD calculations, which speeds each step up, but it does not catch up.

6.2 Netflix Competition Data

We used our `softImpute` package in `R` to fit a sequence of models on the Netflix competition data. Here there are 480,189 users, 17,770 movies and a total of 100,480,507 ratings, making the resulting matrix 98.8% missing. There is a designated test set (the "probe set"), a subset of 1,408,395 of the these ratings, leaving 99,072,112 for training.

Figure 2 compares the performance of `hardImpute` (Mazumder et al., 2010) with `softImpute-ALS` on these data. `hardImpute` uses rank-restricted SVDs iteratively to estimate the missing data, similar to `softImpute` but without shrinkage. The shrinkage helps here, leading to a best test-set RMSE of 0.943. This is a 1% improvement over the "Cinematch" score, somewhat short of the prize-winning improvement of 10%.

Both methods benefit greatly from using warm starts. `hardImpute` is solving a non-convex problem, while the intention is for `softImpute-ALS` to solve the convex problem

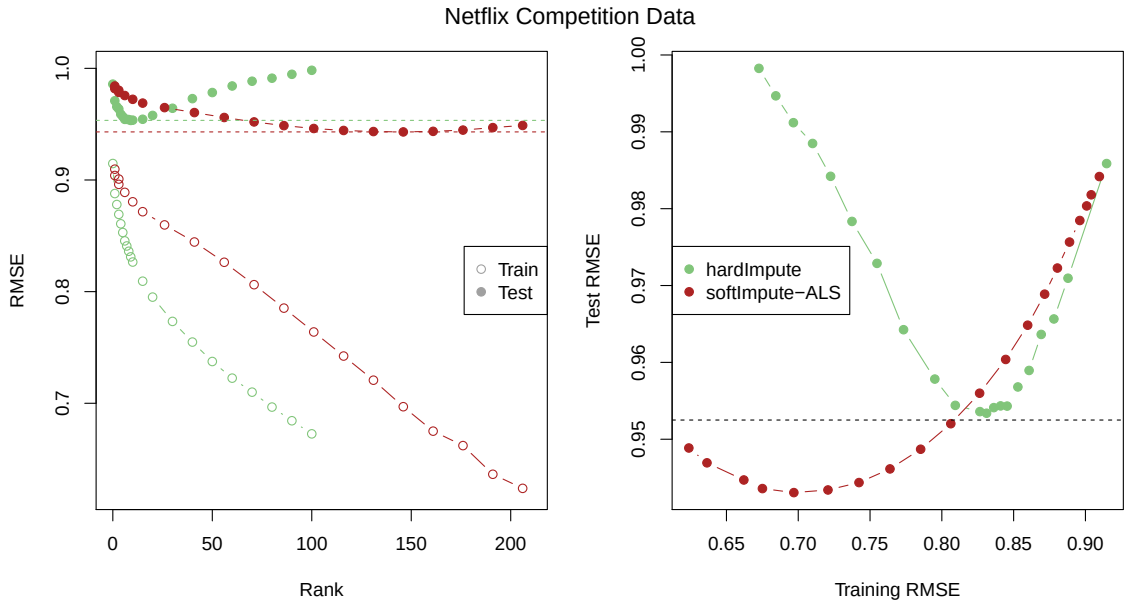


Figure 2: Performance of `hardImpute` versus `softImpute-ALS` on the Netflix data. `hardImpute` uses a rank-restricted SVD at each step of the imputation, while `softImpute-ALS` does shrinking as well. The left panel shows the training and test error as a function of the rank of the solution—an imperfect calibration in light of the shrinkage. The right panel gives the test error as a function of the training error. `hardImpute` fits more aggressively, and overfits far sooner than `softImpute-ALS`. The horizontal dotted line is the "Cinematch" score, the target to beat in this competition.

(1). This will be achieved if the operating rank is sufficiently large. The idea is to decide on a decreasing sequence of values for λ, starting from λ_{max} (the smallest value for which the solution $\widehat{M} = 0$, which corresponds to the largest singular value of $P_\Omega(X)$). Then for each value of λ, use an operating rank somewhat larger than the rank of the previous solution, with the goal of getting the solution rank smaller than the operating rank. The sequence of twenty models took under six hours of computing on a Linux cluster with 300Gb of ram (with a fairly liberal relative convergence criterion of 0.001), using the `softImpute` package in `R`.

Figure 3 (left panel) gives timing comparison results for one of the Netflix fits, this time implemented in Matlab. The right panel gives timing results on the smaller MovieLens 10M matrix. In these applications we need not get a very accurate solution, and so early stopping is an attractive option. `softImpute-ALS` reaches a solution close to the minimum in about 1/4 the time it takes `ALS`.

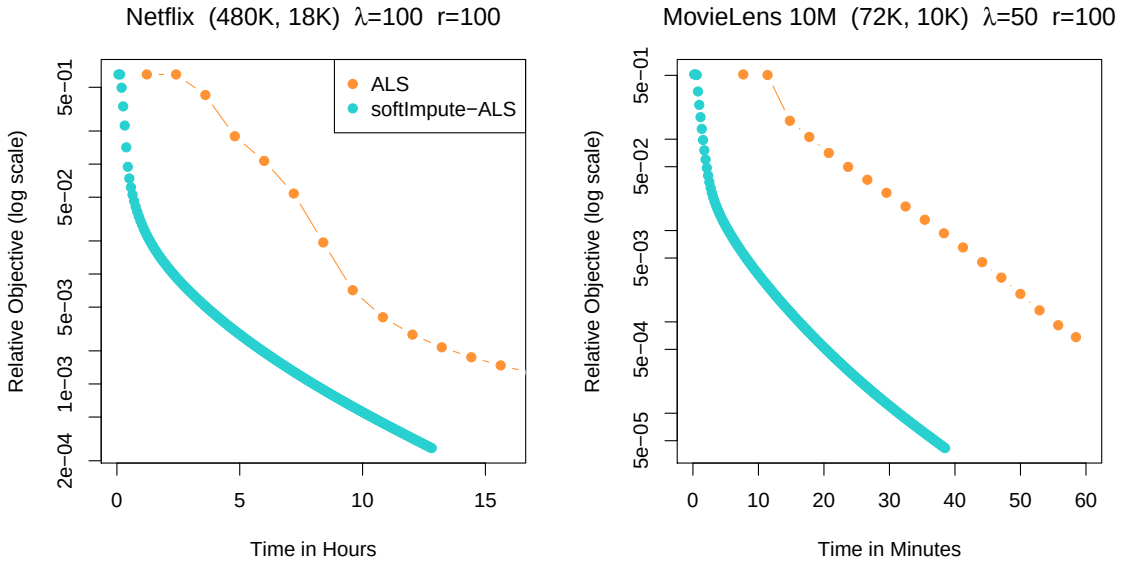


Figure 3: Left: timing results on the Netflix matrix, comparing `ALS` with `softImpute-ALS`. Right: timing on the MovieLens 10M matrix. In both cases we see that while `ALS` makes bigger gains per iteration, each iteration is much more costly.

7. R Package `softImpute`

We have developed an `R` package `softImpute` for fitting these models (Hastie and Mazumder, 2013), which is available on CRAN. The package implements both `softImpute` and `softImpute-ALS`. It can accommodate large matrices if the number of missing entries is correspondingly large, by making use of sparse-matrix formats. There are functions for centering and scaling (see Section 9), and for making predictions from a fitted model. The package also has a function `svd.als` for computing a low-rank SVD of a large sparse matrix, with row and/or column centering. More details can be found in the package Vignette on the first authors web page, at

`http://web.stanford.edu/~hastie/swData/softImpute/vignette.html`.

8. Distributed Implementation

We provide a distributed version of `softimpute-ALS` (given in Algorithm 5.1), built upon the Spark cluster programming framework.

8.1 Design

The input matrix to be factored is split row-by-row across many machines. The transpose of the input is also split row-by-row across the machines. The current model (i.e. the

current guess for A, B) is repeated and held in memory on every machine. Thus the total time taken by the computation is proportional to the number of non-zeros divided by the number of CPU cores, with the restriction that the model should fit in memory.

At every iteration, the current model is broadcast to all machines, such that there is only one copy of the model on each machine. Each CPU core on a machine will process a partition of the input matrix, using the local copy of the model available. This means that even though one machine can have many cores acting on a subset of the input data, all those cores can share the same local copy of the model, thus saving RAM. This saving is especially pronounced on machines with many cores.

The implementation is available online at `http://git.io/sparkfastals` with documentation, in Scala. The implementation has a method named `multByXstar`, corresponding to line 3 of Algorithm 5.1 which multiplies X^* by another matrix on the right, exploiting the "sparse-plus-low-rank" structure of X^*. This method has signature:

```
multByXstar(X: IndexedRowMatrix, A: BDM[Double], B: BDM[Double], C:
BDM[Double])
```

This method has four parameters. The first parameter `X` is a distributed matrix consisting of the input, split row-wise across machines. The full documentation for how this matrix is spread across machines is available online[4]. The `multByXstar` method takes a distributed matrix, along with local matrices `A`, `B`, and `C`, and performs line 3 of Algorithm 5.1 by multiplying X^* by `C`. Similarly, the method `multByXstarTranspose` performs line 5 of Algorithm 5.1.

After each call to `multByXstar`, the machines each will have calculated a portion of A. Once the call finishes, the machines each send their computed portion (which is small and can fit in memory on a single machine, since A can fit in memory on a single machine) to the master node, which will assemble the new guess for A and broadcast it to the worker machines. A similar process happens for `multByXstarTranspose`, and the whole process is repeated every iteration.

8.2 Experiments

We report iteration times using an Amazon EC2 cluster with 10 slaves and one master, of instance type "`c3.4xlarge`". Each machine has 16 CPU cores and 30 GB of RAM. We ran `softimpute-ALS` on matrices of varying sizes with iteration runtimes available in Table 1, setting $k = 5$. Where possible, hardware acceleration was used for local linear algebraic operations, via breeze and BLAS.

The popular Netflix prize matrix has $17,770$ rows, $480,189$ columns, and $100,480,507$ non-zeros. We report results on several larger matrices in Table 1, up to 10 times larger.

9. Centering and Scaling

We often want to remove row and/or column means from a matrix before performing a low-rank SVD or running our matrix completion algorithms. Likewise we may wish to standardize the rows and or columns to have unit variance. In this section we present an

4. `https://spark.apache.org/docs/latest/mllib-basics.html#indexedrowmatrix`

Matrix Size	Number of Nonzeros	Time per iteration (s)
$10^6 \times 10^6$	10^6	5
$10^6 \times 10^6$	10^9	6
$10^7 \times 10^7$	10^9	139

Table 1: Running times for distributed `softimpute-ALS`

algorithm for doing this, in a way that is sensitive to the storage requirement of very large, sparse matrices. We first present our approach, and then discuss implementation details.

We have a two-dimensional array $X = \{X_{ij}\} \in \mathbb{R}^{m\times n}$, with pairs $(i, j) \in \Omega$ observed and the rest missing. The goal is to standardize the rows and columns of X to mean zero and variance one simultaneously. We consider the mean/variance model

$$X_{ij} \sim (\mu_{ij}, \sigma_{ij}^2) \tag{58}$$

with

$$\mu_{ij} = \alpha_i + \beta_j; \tag{59}$$
$$\sigma_{ij} = \tau_i\gamma_j. \tag{60}$$

Given the parameters of this model, we would standardized each observation via

$$\begin{aligned} \tilde{X}_{ij} &= \frac{X_{ij} - \mu_{ij}}{\sigma_{ij}} \\ &= \frac{X_{ij} - \alpha_i - \beta_j}{\tau_i\gamma_j}. \end{aligned} \tag{61}$$

If model (58) were correct, then each entry of the standardized matrix, viewed as a realization of a random variable, would have population mean/variance $(0, 1)$. A consequence would be that realized rows and columns would also have means and variances with expected values zero and one respectively. However, we would like the observed data to have these row and column properties.

Our representation (59)–(60) is not unique, but is easily fixed to be so. We can include a constant μ_0 in (59) and then have α_i and β_j average 0. Likewise, we can have an overall scaling σ_0, and then have $\log \tau_i$ and $\log \gamma_j$ average 0. Since this is not an issue for us, we suppress this refinement.

We are not the first to attempt this dual centering and scaling. Indeed, Olshen and Rajaratnam (2010) implement a very similar algorithm for complete data, and discuss convergence issues. Our algorithm differs in two simple ways: it allows for missing data, and it learns the parameters of the centering/scaling model (61) (rather than just applying them). This latter feature will be important for us in our matrix-completion applications; once we have estimated the missing entries in the standardized matrix $\widetilde{X}$, we will want to *reverse* the centering and scaling on our predictions.

In matrix notation we can write our model

$$\widetilde{\mathbf{X}} = \mathbf{D}_\tau^{-1}(\mathbf{X} - \boldsymbol{\alpha}\mathbf{1}^T - \mathbf{1}\boldsymbol{\beta}^T)\mathbf{D}_\gamma^{-1}, \tag{62}$$

where $\mathbf{D}_\tau = \text{diag}(\tau_1, \tau_2, \ldots, \tau_m)$, similar for $\mathbf{D}_\gamma$, and the missing values are represented in the full matrix as `NA`s (e.g. as in `R`). Although it is not the focus of this paper, this centering model is also useful for large, complete, sparse matrices $\mathbf{X}$ (with many zeros, stored in sparse-matrix format). Centering would destroy the sparsity, but from (62) we can see we can store it in "sparse-plus-low-rank" format. Such a matrix can be left and right-multiplied easily, and hence is ideal for alternating subspace methods for computing a low-rank SVD. The function `svd.als` in the `softImpute` package (section 7) can accommodate such structure.

9.1 Method-of-moments Algorithm

We now present an algorithm for estimating the parameters. The idea is to write down four systems of estimating equations that demand that the transformed observed data have row means zero and variances one, and likewise for the columns. We then iteratively solve these equations, until all four conditions are satisfied simultaneously. We do not in general have any guarantees that this algorithm will always converge except in the noted special cases, but empirically we typically see rapid convergence.

Consider the estimating equation for the row-means condition (for each row i)

$$\begin{aligned} \frac{1}{n_i}\sum_{j\in\Omega_i}\widetilde{X}_{ij} &= \frac{1}{n_i}\sum_{j\in\Omega_i}\frac{X_{ij}-\alpha_i-\beta_j}{\tau_i\gamma_j} \\ &= 0, \end{aligned} \tag{63}$$

where $\Omega_i = \{j|(i,j)\in\Omega\}$, and $n_i = |\Omega_i| \le n$. Rearranging we get

$$\alpha_i = \frac{\sum_{j\in\Omega_i}\frac{1}{\gamma_j}(X_{ij}-\beta_j)}{\sum_{j\in\Omega_i}\frac{1}{\gamma_j}}, \quad i=1,\ldots,m. \tag{64}$$

This is a weighted mean of the partial residuals $X_{ij}-\beta_j$ with weights inversely proportional to the column standard-deviation parameters γ_j. By symmetry, we get a similar equation for β_j,

$$\beta_j = \frac{\sum_{i\in\Omega^j}\frac{1}{\tau_i}(X_{ij}-\alpha_i)}{\sum_{i\in\Omega^j}\frac{1}{\tau_i}}, \quad j=1,\ldots,n, \tag{65}$$

where $\Omega^j = \{i|(i,j)\in\Omega\}$, and $m_j = |\Omega^j| \le m$.

Similarly, the variance conditions for the rows are

$$\begin{aligned} \frac{1}{n_i}\sum_{j\in\Omega_i}\widetilde{X}_{ij}^2 &= \frac{1}{n_i}\sum_{j\in\Omega_i}\frac{(X_{ij}-\alpha_i-\beta_j)^2}{\tau_i^2\gamma_j^2} \\ &= 1, \end{aligned} \tag{66}$$

which simply says

$$\tau_i^2 = \frac{1}{n_i}\sum_{j\in\Omega_i}\frac{(X_{ij}-\alpha_i-\beta_j)^2}{\gamma_j^2}, \quad i=1,\ldots,m. \tag{67}$$

Likewise

$$\gamma_j^2 = \frac{1}{m_j}\sum_{i\in\Omega^j}\frac{(X_{ij}-\alpha_i-\beta_j)^2}{\tau_i^2}, \quad j=1,\ldots,n. \tag{68}$$

The method-of-moments estimators require iterating these four sets of equations (64), (65), (67), (68) till convergence. We monitor the following functions of the "residuals"

$$R = \sum_{i=1}^{m}\left[\frac{1}{n_i}\sum_{j\in\Omega_i}\widetilde{X}_{ij}\right]^2+\sum_{j=1}^{n}\left[\frac{1}{m_j}\sum_{i\in\Omega^j}\widetilde{X}_{ij}\right]^2 \tag{69}$$

$$+\sum_{i=1}^{m}\log^2\left(\frac{1}{n_i}\sum_{j\in\Omega_i}\widetilde{X}_{ij}^2\right)+\sum_{j=1}^{n}\log^2\left(\frac{1}{m_j}\sum_{i\in\Omega^j}\widetilde{X}_{ij}^2\right) \tag{70}$$

In experiments it appears that R converges to zero very fast, perhaps linear convergence. In Appendix B we show slightly different versions of these estimators which are more suitable for sparse-matrix calculations.

In practice we may not wish to apply all four standardizations, but instead a subset. For example, we may wish to only standardize columns to have mean zero and variance one. In this case we simply set the omitted centering parameters to zero, and scaling parameters to one, and skip their steps in the iterative algorithm. In certain cases we have convergence guarantees:

- Column-only centering and/or scaling. Here no iteration is required; the centering step precedes the scaling step, and we are done. Likewise for row-only.
- Centering only, no scaling. Here the situation is exactly that of an unbalanced two-way ANOVA, and our algorithm is exactly the Gauss-Seidel algorithm for fitting the two-way ANOVA model. This is known to converge, modulo certain degenerate situations.

For the other cases we have no guarantees of convergence.

We present an alternative sequence of formulas in Appendix B which allows one to simultaneously apply the transformations, and learn the parameters.

10. Discussion

We have presented a new algorithm for matrix completion, suitable for solving Problem (1) for very large problems, as long as the solution rank is manageably low. Our algorithm capitalizes on the different strengths and weakness in each of the popular alternatives:

- `ALS` has to solve a different regression problem for every row/column, because of their different amount of missingness, and this can be costly. `softImpute-ALS` solves a single regression problem once and simultaneously for all the rows/columns, because it operates on a filled-in matrix which is complete. Although these steps are typically not as strong as those of `ALS`, the speed advantage more than compensates.
- `softImpute` wastes time in early iterations computing a low-rank SVD of a far-from-optimal estimate, in order to make its next imputation. One can think of `softImpute-ALS` as simultaneously filling in the matrix at each alternating step, as it is computing the SVD. By the time it is done, it has the the solution sought by `softImpute`, but with far fewer iterations.

`softImpute` allows for an extremely efficient distributed implementation (Section 8), and hence can scale to large problems, given a sufficiently large computing infrastructure.

Acknowledgments

The authors thank Balasubramanian Narasimhan for helpful discussions on distributed computing in R. The first author thanks Andreas Buja and Stephen Boyd for stimulating "footnote" discussions that led to the centering/scaling in Section 9. Trevor Hastie was partially supported by grant DMS-1407548 from the National Science Foundation, and grant RO1-EB001988-15 from the National Institutes of Health. Rahul Mazumder was funded in part by Columbia University's start-up funds and a grant from the Betty-Moore Sloan Foundation.

Appendix A. Proofs from Section 5.1

Here, we gather some proofs and technical details from Section 5.1.

A.1 Proof of Lemma 2

To prove this we begin with the following elementary result concerning a ridge regression problem:

Lemma 5 *Consider a ridge regression problem*

$$H(\beta) := \tfrac{1}{2}\|y - M\beta\|_2^2 + \frac{\lambda}{2}\|\beta\|_2^2 \tag{71}$$

with $\beta^* \in \arg\min_\beta\ H(\beta)$. *Then the following inequality is true:*

$$H(\beta) - H(\beta^*) = \frac{1}{2}(\beta - \beta^*)^T(M^TM + \lambda\mathbf{I})(\beta - \beta^*) = \tfrac{1}{2}\|M(\beta - \beta^*)\|_2^2 + \frac{\lambda}{2}\|\beta - \beta^*\|_2^2$$

Proof *The proof follows from the second order Taylor Series expansion of* $H(\beta)$*:*

$$H(\beta) = H(\beta^*) + \langle \nabla H(\beta^*), \beta - \beta^* \rangle + \frac{1}{2}(\beta - \beta^*)^T(M^TM + \lambda\mathbf{I})(\beta - \beta^*)$$

and observing that $\nabla H(\beta^*) = 0$. ■

We will need to obtain a lower bound on the difference $F(A_{k+1}, B_k) - F(A_k, B_k)$. Towards this end we make note of the following chain of inequalities:

$$\begin{aligned} F(A_k, B_k) &= g(A_k B_k^T) + \frac{\lambda}{2}(\|A_k\|_F^2 + \|B_k\|_F^2) && (72)\\ &= Q_A(A_k|A_k, B_k) && (73)\\ &\geq \min_{Z_1} \; Q_A(Z_1|A_k, B_k) && (74)\\ &= Q_A(A_{k+1}|A_k, B_k) && (75)\\ &\geq g(A_{k+1}B_k^T) + \frac{\lambda}{2}(\|A_{k+1}\|_F^2 + \|B_k\|_F^2) && (76)\\ &= F(A_{k+1}, B_k) && (77) \end{aligned}$$

where, Line (73) follows from (31), and (76) follows from (30).

Clearly, from Lines (77) and (72) we have (78)

$$\begin{aligned} F(A_k, B_k) - F(A_{k+1}, B_k) &\geq Q_A(A_k|A_k, B_k) - Q_A(A_{k+1}|A_k, B_k) && (78)\\ &= \tfrac{1}{2}\|(A_{k+1} - A_k)B_k^T\|_2^2 + \frac{\lambda}{2}\|A_{k+1} - A_k\|_2^2, && (79) \end{aligned}$$

where, (79) follows from (78) using Lemma 5.

Similarly, following the above steps for the B-update we have:

$$F(A_k, B_k) - F(A_{k+1}, B_{k+1}) \geq \tfrac{1}{2}\|A_{k+1}(B_{k+1} - B_k)^T\|_2^2 + \frac{\lambda}{2}\|B_{k+1} - B_k\|_2^2. \quad (80)$$

Adding (79) and (80) we get (35) concluding the proof of the lemma.

A.2 Proof of Lemma 3

Let us use the shorthand Δ in place of $\Delta\left((A, B), (A^+, B^+)\right)$ as defined in (37).

First of all observe that the result (35) can be replaced with $(A_k, B_k) \leftarrow (A, B)$ and $(A_{k+1}, B_{k+1}) \leftarrow (A^+, B^+)$. This leads to the following:

$$\begin{aligned} F(A, B) - F(A^+, B^+) \geq \tfrac{1}{2}\left(\|(A - A^+)B^T\|_F^2 + \|A^+(B^+ - B)^T\|_F^2\right) \\ + \frac{\lambda}{2}\left(\|A - A^+\|_F^2 + \|B^+ - B\|_F^2\right). \end{aligned} \quad (81)$$

First of all, it is clear that if A, B is a fixed point then $\Delta = 0$.

Let us consider the converse, i.e., the case when $\Delta = 0$. Note that if $\Delta = 0$ then each of the summands appearing in the definition of Δ is also zero. We will now make use of the interesting result (that follows from the Proof of Lemma 2) in (78) and (79) which says:

$$Q_A(A|A, B) - Q_A(A^+|A, B) = \tfrac{1}{2}\|(A^+ - A)B^T\|_2^2 + \frac{\lambda}{2}\|A^+ - A\|_2^2.$$

Now the right hand side of the above equation is zero (since $\Delta = 0$) which implies that, $Q_A(A|A, B) - Q_A(A^+|A, B) = 0$. An analogous result holds true for B.

Using the nesting property (34), it follows that $F(A, B) = F(A_+, B_+)$—thereby showing that (A, B) is a fixed point of the algorithm.

A.3 Proof of Theorem 4

We make use of (35) and add both sides of the inequality over $k = 1, \ldots, K$, which leads to:

$$\sum_{i=1}^{K} \left(F(A_k, B_k) - F(A_{k+1}, B_{k+1})\right) \geq \sum_{k=1}^{K} \eta_k \geq K(\min_{K \geq k \geq 1} \eta_k) \tag{82}$$

Since, $F(A_k, B_k)$ is a decreasing sequence (bounded below) it converges to F^{∞} say. It follows that:

$$\begin{aligned} \sum_{i=1}^{K} \left(F(A_k, B_k) - F(A_{k+1}, B_{k+1})\right) = &\quad F(A^1, B^1) - F(A^{K+1}, B^{K+1}) \\ \leq &\quad F(A^1, B^1) - F^{\infty} \end{aligned} \tag{83}$$

Using (83) along with (82) we have the following convergence rate:

$$\min_{1 \leq k \leq K} \eta_k \leq \left(F(A^1, B^1) - F(A^{\infty}, B^{\infty})\right)/K,$$

thereby completing the proof of the theorem.

A.4 Proof of Corollary 1

Recall the definition of η_k

$$\eta_k = \tfrac{1}{2}\left(\|(A_k - A_{k+1})B_k^T\|_F^2 + \|A_{k+1}(B_k - B_{k+1})^T\|_F^2\right) + \frac{\lambda}{2}\left(\|A_k - A_{k+1}\|_F^2 + \|B_k - B_{k+1}\|_F^2\right)$$

Since we have assumed that

$$\ell^U \mathbf{I} \succeq B_k^T B_k \succeq \ell^L \mathbf{I}, \;\; \ell^U \mathbf{I} \succeq A_k^T A_k \succeq \ell^L \mathbf{I}, \quad \forall k$$

we then have:

$$\eta_k \geq \left(\frac{\ell^L}{2} + \frac{\lambda}{2}\right)\|A_k - A_{k+1}\|_F^2 + \left(\frac{\ell^L}{2} + \frac{\lambda}{2}\right)\|B_k - B_{k+1}\|_F^2.$$

Using the above in (82) and assuming that $\ell_L > 0$, we have the bound:

$$\min_{1 \leq k \leq K} \left(\|(A_k - A_{k+1})\|_F^2 + \|B_k - B_{k+1}\|_F^2\right) \leq \frac{2}{(\ell^L + \lambda)}\left(\frac{F(A^1, B^1) - F^{\infty}}{K}\right) \tag{84}$$

Suppose instead of the proximity measure:

$$\left(\|(A_k - A_{k+1})\|_F^2 + \|B_k - B_{k+1}\|_F^2\right),$$

we use the proximity measure:

$$\left(\|(A_k - A_{k+1})B_k^T\|_F^2 + \|A_{k+1}(B_k - B_{k+1})\|_F^2\right).$$

Then observing that:

$$\ell^U\|(A_k - A_{k+1})\|_F^2 \geq \|(A_k - A_{k+1})B_k^T\|_F^2, \quad \ell^U\|B_k - B_{k+1}\|_F^2 \geq \|A_{k+1}(B_k - B_{k+1})^T\|_F^2$$

we have:

$$\eta_k \geq \left(\frac{\lambda}{2\ell^U} + \frac{1}{2}\right)\left(\|(A_k - A_{k+1})B_k^T\|_F^2 + \|A_{k+1}(B_k - B_{k+1})\|_F^2\right).$$

Using the above bound in (82) we arrive at a bound which is similar in spirit to (41) but with a different proximity measure on the step-sizes:

$$\min_{1\leq k\leq K}\left(\|(A_k - A_{k+1})B_k^T\|_F^2 + \|A_{k+1}(B_k - B_{k+1})\|_F^2\right) \leq \frac{2\ell^U}{\lambda + \ell_U}\left(\frac{F(A^1, B^1) - F^\infty}{K}\right) \quad (85)$$

It is useful to contrast results (41) and (42) with the case $\lambda = 0$.

$$\min_{1\leq k\leq K}\left(\|(A_k - A_{k+1})B_k^T\|_F^2 + \|A_{k+1}(B_k - B_{k+1})\|_F^2\right) \leq \begin{cases} \frac{2\ell^U}{\lambda+\ell_U}\left(\frac{F(A^1,B^1)-F^\infty}{K}\right) & \lambda > 0 \\ 2\ell^U\left(\frac{F(A^1,B^1)-F^\infty}{K}\right) & \lambda = 0 \end{cases} \quad (86)$$

The convergence rate with the other proximity measure on the step-sizes have the following two cases:

$$\min_{1\leq k\leq K}\left(\|(A_k - A_{k+1})\|_F^2 + \|B_k - B_{k+1}\|_F^2\right) \leq \begin{cases} \frac{2}{(\ell^L+\lambda)}\left(\frac{F(A^1,B^1)-F^\infty}{K}\right) & \lambda > 0, \\ \frac{2}{\ell^L}\left(\frac{F(A^1,B^1)-F^\infty}{K}\right) & \lambda = 0. \end{cases} \quad (87)$$

The assumption (40) $\ell^U\mathbf{I} \succeq B_k^T B_k$ and $\ell^U\mathbf{I} \succeq A_k^T A_k$ can be interpreted as an upper bounds to the locally Lipschitz constants of the gradients of $Q_A(Z|A_k, B_k)$ and $Q_B(Z|A_{k+1}, B_k)$ for all k:

$$\begin{aligned} \|\nabla Q_A(A_{k+1}|A_k, B_k) - \nabla Q_A(A_k|A_k, B_k)\| &\leq \ell^U\|A_{k+1} - A_k\|, \\ \|\nabla Q_B(B_k|A_{k+1}, B_k) - \nabla Q_B(B_{k+1}|A_{k+1}, B_k)\| &\leq \ell^U\|B_{k+1} - B_k\|. \end{aligned} \quad (88)$$

The above leads to convergence rate bounds on the (partial) gradients of the function $F(A, B)$, i.e.,

$$\min_{1\leq k\leq K}\left(\|\nabla_A f(A_k, B_k)\|^2 + \|\nabla_B f(A_{k+1}, B_k)\|^2\right) \leq \frac{2(\ell^U)^2}{(\ell^L + \lambda)}\left(\frac{F(A^1, B^1) - F^\infty}{K}\right)$$

A.5 Proof of Theorem 5

Proof Part (a):
We make use of the convergence rate derived in Theorem 4. As $k \to \infty$, it follows that $\eta_k \to 0$. This describes the fate of the objective values $F(A_k, B_k)$, but does not inform us about the properties of the sequence A_k, B_k. Towards this end, note that if $\lambda > 0$, then the sequence A_k, B_k is bounded and thus has a limit point. Let A_*, B_* be any limit point of the sequence A_k, B_k. It follows by a simple subsequence argument that $F(A_k, B_k) \to F(A_*, B_*)$ and A_*, B_* is a fixed point of Algorithm 5.1 and in particular a first order stationary point of Problem (6).
Part (b):
The sequence (A_k, B_k) need not have a unique limit point, however, we show below: for every subsequence of B_k that converges, the corresponding subsequence of A_k also converges.

Suppose, $B_k \to B_*$ (along a subsequence $k \in \nu$). We will show that the sequence A_k for $k \in \nu$ has a unique limit point.

We argue by the method of contradiction. Suppose there are two limit points of $A_k, k \in \nu$, namely, A_1 and A_2 and $A_{k_1} \to A_1, k_1 \in \nu_1 \subset \nu$ and $A_{k_2} \to A_2, k_2 \in \nu_2 \subset \nu$ with $A_1 \neq A_2$.

Consider the objective value sequence: $F(A_k, B_k)$. For fixed B_k the update in A from A_k to A_{k+1} results in

$$F(A_k, B_k) - F(A_{k+1}, B_k) \geq \frac{\lambda}{2}\|A_k - A_{k+1}\|_F^2.$$

Take $k_1 \in \nu_1$ and $k_2 \in \nu_2$, we have:

$$\begin{aligned} F(A_{k_2}, B_{k_2}) - F(A_{k_1+1}, B_{k_1}) &= (F(A_{k_2}, B_{k_2}) - F(A_{k_2}, B_{k_1})) \\ &\quad + (F(A_{k_2}, B_{k_1}) - F(A_{k_1+1}, B_{k_1})) \qquad (89) \\ &\geq (F(A_{k_2}, B_{k_2}) - F(A_{k_2}, B_{k_1})) + \frac{\lambda}{2}\|A_{k_2} - A_{k_1+1}\|_F^2 \qquad (90) \end{aligned}$$

where Line 90 follows by using Lemma 5. As $k_1, k_2 \to \infty$, $B_{k_2}, B_{k_1} \to B_*$ hence,

$$F(A_{k_2}, B_{k_2}) - F(A_{k_2}, B_{k_1}) \to 0, \text{ and } \|A_{k_2} - A_{k_1+1}\|_F^2 \to \|A_2 - A_1\|_F^2$$

However, the lhs of (89) converges to zero, which is a contradiction. This implies that $\|A_2 - A_1\|_F^2 = 0$ i.e. A_k for $k \in \nu$ has a unique limit point.

Exactly the same argument holds true for the sequence A_k, leading to the conclusion of the other part of Part (b). ■

Appendix B. Alternative Computing Formulas for Method of Moments

In this section we present the same algorithm, but use a slightly different representation. For matrix-completion problems, this does not make much of a difference in terms of computational load. But we also have other applications in mind, where the large matrix X may be fully observed, but is very sparse. In this case we do not want to actually apply the centering operations; instead we represent the matrix as a "sparse-plus-low-rank" object, a class for which we have methods for simple row and column operations.

Consider the row-means (for each row i). We can introduce a change Δ_i^α from the old α_i^o to the new α_i. Then we have

$$\begin{aligned} \sum_{j \in \Omega_i} \widetilde{X}_{ij} &= \sum_{j \in \Omega_i} \frac{X_{ij} - \alpha_i^o - \Delta_i^\alpha - \beta_j}{\tau_i \gamma_j} \qquad (91) \\ &= 0, \end{aligned}$$

where as before $\Omega_i = \{j | (i,j) \in \Omega\}$. Rearranging we get

$$\Delta_i^\alpha = \frac{\sum_{j \in \Omega_i} \widetilde{X}_{ij}^o}{\sum_{j \in \Omega_i} \frac{1}{\tau_i \gamma_j}}, \quad i = 1, \ldots, m, \qquad (92)$$

where

$$\widetilde{X}_{ij}^{o} = \frac{X_{ij} - \alpha_i^o - \beta_j}{\tau_i \gamma_j}. \tag{93}$$

Then $\alpha_i = \alpha_i^o + \Delta_i^\alpha$. By symmetry, we get a similar equation for Δ_j^β,

Likewise for the variances.

$$\begin{aligned} \frac{1}{n_i} \sum_{j \in \Omega_i} \widetilde{X}_{ij}^2 &= \frac{1}{n_i} \sum_{j \in \Omega_i} \frac{(X_{ij} - \alpha_i - \beta_j)^2}{(\tau_i \Delta_i^\tau)^2 \gamma_j^2} && (94) \\ &= \frac{1}{n_i} \sum_{j \in \Omega_i} \left(\frac{\widetilde{X}_{ij}^o}{\Delta_i^\tau} \right)^2 && (95) \\ &= 1. \end{aligned}$$

Here we modify τ_i by a multiplicative factor Δ_i^τ. Here the solution is

$$(\Delta_i^\tau)^2 = \frac{1}{n_i} \sum_{j \in \Omega_i} (\widetilde{X}_{ij}^o)^2, \quad i = 1, \ldots, m. \tag{96}$$

By symmetry, we get a similar equation for Δ_j^γ,

The method-of-moments estimators amount to iterating these four sets of equations till convergence. Now we can monitor the changes via

$$R = \sum_{i=1}^{m} {\Delta_i^\alpha}^2 + \sum_{j=1}^{n} {\Delta_j^\beta}^2 + \sum_{i=1}^{m} \log^2 \Delta_i^\tau + \sum_{j=1}^{n} \log^2 \Delta_j^\gamma \tag{97}$$

which should converge to zero.

References

Dimitri P Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, Massachusetts, 2nd edition, 1999. ISBN 1886529000. URL `http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/1886529000`.

Samuel Burer and Renato D.C. Monteiro. Local minima and convergence in low-rank semidefinite programming. *Mathematical Programming*, 103(3):427–631, 2005.

Emmanuel Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 2008. doi: 10.1007/s10208-009-9045-5. URL `http://dx.doi.org/10.1007/s10208-009-9045-5`.

Emmanuel J. Candès and Terence Tao. The power of convex relaxation: Near-optimal matrix completion, 2009. URL `http://www.citebase.org/abstract?id=oai:arXiv.org:0903.1476`.

Caihua Chen, Bingsheng He, and Xiaoming Yuan. Matrix completion via an alternating direction method. *IMA Journal of Numerical Analysis*, 32(1):227–245, 2012.

G. Golub and C. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 3 edition, 2012.

Moritz Hardt. Understanding alternating minimization for matrix completion. In *Foundations of Computer Science (FOCS), 2014 IEEE 55th Annual Symposium on*, pages 651–660. IEEE, 2014.

Trevor Hastie and Rahul Mazumder. *softImpute: Matrix Completion via Iterative Soft-Thresholded Svd*, 2013. URL `http://CRAN.R-project.org/package=softImpute`. R package version 1.0.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning, Second Edition: Data Mining, Inference, and Prediction (Springer Series in Statistics)*. Springer New York, 2 edition, 2009. ISBN 0387848576.

Prateek Jain, Praneeth Netrapalli, and Sujay Sanghavi. Low-rank matrix completion using alternating minimization. In *Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing*, pages 665–674. ACM, 2013.

Michel Journée, F Bach, P-A Absil, and Rodolphe Sepulchre. Low-rank optimization on the cone of positive semidefinite matrices. *SIAM Journal on Optimization*, 20(5):2327–2351, 2010.

Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

R.M. Larsen. Propack-software for large and sparse svd calculations, 2004. URL `http://sun.stanford.edu/~rmunk/PROPACK/`.

A. Lewis. Derivatives of spectral functions. *Mathematics of Operations Research*, 21(3): 576–588, 1996.

Rahul Mazumder, Trevor Hastie, and Rob Tibshirani. Spectral regularization algorithms for learning large incomplete matrices. *Journal of Machine Learning Research*, 11:2287–2322, 2010.

Leon Mirsky. A trace inequality of John von Neumann. *Monatshefte für Mathematik*, 79 (4):303–306, 1975.

Richard Olshen and Bala Rajaratnam. Successive normalization of rectangular arrays. *Annals of Statistics*, 38(3):1638–1664, 2010.

Meisam Razaviyayn, Mingyi Hong, and Zhi-Quan Luo. A unified convergence analysis of block successive minimization methods for nonsmooth optimization. *SIAM Journal on Optimization*, 23(2):1126–1153, 2013.

J. Rennie and N. Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *ICML*, 2005.

Nathan Srebro, Jason Rennie, and Tommi Jaakkola. Maximum margin matrix factorization. *Advances in Neural Information Processing Systems*, 17, 2005.

G. Stewart and Ji-Guang Sun. *Matrix Perturbation Theory*. Academic Press, Boston, 1 edition, 1990. ISBN 0126702306. URL http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0126702306.

Yunhong Zhou, Dennis Wilkinson, Robert Schreiber, and Rong Pan. Large-scale parallel collaborative filtering for the netflix prize. In *Algorithmic Aspects in Information and Management*, pages 337–348. Springer, 2008.

Journal of Machine Learning Research 16 (2015) 3403-3454 Submitted 12/14; Revised 4/15; Published 12/15

On the Inductive Bias of Dropout

David P. Helmbold DPH@SOE.UCSC.EDU
Department of Computer Science
University of California, Santa Cruz
Santa Cruz, CA 95064, USA

Philip M. Long PLONG@MICROSOFT.COM
Microsoft
1020 Enterprise Way
Sunnyvale, CA 94089, USA

Editor: Samy Bengio

Abstract

Dropout is a simple but effective technique for learning in neural networks and other settings. A sound theoretical understanding of dropout is needed to determine when dropout should be applied and how to use it most effectively. In this paper we continue the exploration of dropout as a regularizer pioneered by Wager et al. We focus on linear classification where a convex proxy to the misclassification loss (i.e. the logistic loss used in logistic regression) is minimized. We show:

- when the dropout-regularized criterion has a unique minimizer,
- when the dropout-regularization penalty goes to infinity with the weights, and when it remains bounded,
- that the dropout regularization can be non-monotonic as individual weights increase from 0, and
- that the dropout regularization penalty may *not* be convex.

This last point is particularly surprising because the combination of dropout regularization with any convex loss proxy is always a convex function.

In order to contrast dropout regularization with L_2 regularization, we formalize the notion of when different random sources of data are more compatible with different regularizers. We then exhibit distributions that are provably more compatible with dropout regularization than L_2 regularization, and vice versa. These sources provide additional insight into how the inductive biases of dropout and L_2 regularization differ. We provide some similar results for L_1 regularization.

Keywords: dropout, inductive bias, learning theory, regularization, feature noising

1. Introduction

Since its prominent role in a win of the ImageNet Large Scale Visual Recognition Challenge (Hinton, 2012; Hinton et al., 2012; Srivastava et al., 2014), there has been intense interest in dropout (see the work by Dahl, 2012; Deng et al., 2013; Dahl et al., 2013; Wan et al., 2013; Wager et al., 2013; Baldi and Sadowski, 2014; Van Erven et al., 2014). Dropout is a modification of stochastic gradient descent where each update is performed on a reduced

network created by temporarily removing a random subset of the nodes. This paper studies the inductive bias of dropout: when one chooses to train with dropout, what prior preference over models results? We show that dropout training shapes the learner's search space in a much different way than L_1 or L_2 regularization. Our results shed new insight into why dropout prefers rare features, how the dropout probability affects the strength of regularization, and how dropout restricts the co-adaptation of weights.

Our theoretical study will concern learning a linear classifier via convex optimization. The learner wishes to find a parameter vector $\mathbf{w}$ so that, for a random feature-label pair $(\mathbf{x}, y) \in \mathbf{R}^n \times \{-1, 1\}$ drawn from some joint distribution P, the probability that $\text{sign}(\mathbf{w} \cdot \mathbf{x}) \neq y$ is small. It does this by using training data to try to minimize $\mathbf{E}(\ell(y\mathbf{w} \cdot \mathbf{x}))$, where $\ell(z) = \ln(1 + \exp(-z))$ is the loss function associated with logistic regression.

We have chosen to focus on this problem for several reasons. First, the inductive bias of dropout is not well understood even in this simple setting. Second, linear classifiers remain a popular choice for practical problems, especially in the case of very high-dimensional data. Third, we view a thorough understanding of dropout in this setting as a mandatory prerequisite to understanding the inductive bias of dropout when applied in a deep learning architecture. This is especially true when the preference over deep learning models is decomposed into preferences at each node. In any case, the setting that we are studying faithfully describes the inductive bias of a deep learning system at its output nodes.

We will borrow the following clean and illuminating description of dropout as artificial noise due to Wager et al. (2013). An algorithm for linear classification using loss ℓ and dropout updates its parameter vector $\mathbf{w}$ online, using stochastic gradient descent. Given an example $(\mathbf{x}, y)$, the dropout algorithm independently perturbs each feature i of $\mathbf{x}$: with probability q, x_i is replaced with 0, and, with probability $p = 1 - q$, x_i is replaced with x_i/p. Equivalently, $\mathbf{x}$ is replaced by $\mathbf{x} + \boldsymbol{\nu}$, where

$$\nu_i = \begin{cases} -x_i & \text{with probability } q \\ (1/p - 1)x_i & \text{with probability } p = 1 - q \end{cases}$$

before performing the stochastic gradient update step. (Note that, while $\boldsymbol{\nu}$ obviously depends on $\mathbf{x}$, if we sample the components of $\mathbf{b} \in \{-1, 1/p-1\}^n$ independently of one another and $\mathbf{x}$, by choosing $b_i = -1$ with the dropout probability q, then we may write $\nu_i = b_i x_i$.)

Stochastic gradient descent is known to converge under a broad variety of conditions (Kushner and Yin, 1997). Thus, if we abstract away sampling issues as done by Breiman (2004); Zhang (2004); Bartlett et al. (2006); Long and Servedio (2010), we are led to consider

$$\mathbf{w}^* \stackrel{\text{def}}{=} \text{argmin}_{\mathbf{w}} \mathbf{E}_{(\mathbf{x},y)\sim P,\boldsymbol{\nu}}(\ell(y\mathbf{w} \cdot (\mathbf{x} + \boldsymbol{\nu})))$$

as dropout can be viewed as a stochastic gradient update of this global objective function. We call this objective the *dropout criterion*, and it can be viewed as a risk on the dropout-induced distribution. (Abstracting away sampling issues is consistent with our goal of concentrating on the inductive bias of the algorithm. From the point of view of a bias-variance decomposition, we do not intend to focus on the large-sample-size case, where the variance is small, but rather to focus on the contribution from the bias where P could be an empirical sample distribution.)

We start with the observation of Wager et al. (2013) that the dropout criterion may be decomposed as

$$\mathbf{E}_{(\mathbf{x},y)\sim P,\boldsymbol{\nu}}(\ell(y\mathbf{w}\cdot(\mathbf{x}+\boldsymbol{\nu}))) = \mathbf{E}_{(\mathbf{x},y)\sim P}(\ell(y\mathbf{w}\cdot\mathbf{x})) + \mathbf{reg}_{D,q}(\mathbf{w}), \tag{1}$$

where $\mathbf{reg}_{D,q}(\mathbf{w})$ is non-negative, and depends only on the marginal distribution D over the feature vectors $\mathbf{x}$ (along with the dropout probability q), and not on the labels. This leads naturally to a view of dropout as a regularizer.

A popular style of learning algorithm minimizes an objective function like the RHS of (1), but where $\mathbf{reg}_{D,q}(\mathbf{w})$ is replaced by a norm of $\mathbf{w}$. One motivation for algorithms in this family is to first replace the training error with a convex proxy to make optimization tractable, and then to regularize using a convex penalty such as a norm, so that the objective function remains convex.

We show that $\mathbf{reg}_{D,q}(\mathbf{w})$ formalizes a preference for classifiers that assign a very large weight to a single feature. This preference is stronger than what one gets from a penalty proportional to $||\mathbf{w}||_1$. In fact, despite the convexity of the dropout risk, we show that $\mathbf{reg}_{D,q}(\mathbf{w})$ is *not* convex. Therefore that dropout provides a way to realize the inductive bias arising from a non-convex penalty while still enjoying the benefit of convexity in the overall objective function (see the plots in Figures 1, 2 and 3). Figure 1 shows the even more surprising result that the dropout regularization penalty is not even monotonic in the absolute values of the individual weights.

It is not hard to see that $\mathbf{reg}_{D,q}(\mathbf{0}) = 0$. Thus, if $\mathbf{reg}_{D,q}(\mathbf{w})$ is greater than the expected loss incurred by $\mathbf{0}$ (which is $\ln 2$), then it might as well be infinity, because dropout will prefer $\mathbf{0}$ to $\mathbf{w}$. However, in some cases, dropout never reaches this extreme—it remains willing to use a models with arbitrarily large parameters, unlike methods that use a convex penalty. In particular,

$$\mathbf{reg}_{D,q}(w_1, 0, 0, 0, ..., 0) < \ln 2$$

for all D, no matter how large w_1 gets. On the other hand, except for some special cases (which are detailed in the body of the paper),

$$\mathbf{reg}_{D,q}(cw_1, cw_2, 0, 0, ..., 0)$$

goes to infinity with c. It follows that $\mathbf{reg}_{D,q}(\mathbf{w})$ cannot be approximated to within any factor, constant or otherwise, by a convex function of $\mathbf{w}$.

To get a sense of which sources dropout can be successfully applied to, we compare dropout with an algorithm that regularizes using L_2, by minimizing the L_2 *criterion*:

$$\mathbf{E}_{(\mathbf{x},y)\sim P}(\ell(y\mathbf{w}\cdot\mathbf{x})) + \frac{\lambda}{2}||\mathbf{w}||_2^2. \tag{2}$$

Will will use "L_2" as a shorthand to refer to an algorithm that minimizes (2). Note that q, the probability of dropping out an input feature, plays a role in dropout analogous to λ. In particular, as q goes to zero the examples remain unperturbed and the dropout regularization has no effect.

Informally, we say that joint probability distributions P and Q *separate* dropout from L_2 if, when the same parameters λ and q are used for both P and Q, then using dropout leads to a much more accurate hypothesis for P, and using L_2 leads to a much more accurate

hypothesis for Q. This enables us to illustrate the inductive biases of the algorithms through contrasting sources that either align or are incompatible with the algorithms' inductive bias. Comparing with another regularizer helps to restrict these illustrative examples to "reasonable" sources, which can be handled using the other regularizer. Ensuring that the same values of the regularization parameter are used for both P and Q controls for the amount of regularization, and ensures that the difference is due to the model preferences of the respective regularizers. This style of analysis is new, as far as we know, and may be a useful tool for studying the inductive biases of other algorithms and in other settings.

Related previous work. Our research builds on the work of Wager et al. (2013), who analyzed dropout for random (x, y) pairs where the distribution of y given x comes from a member of the exponential family, and the quality of a model is evaluated using the log-loss. They pointed out that, in these cases, the dropout criterion can be decomposed into the original loss and a term that does not depend on y, which therefore can be viewed as a regularizer. They then proposed an approximation to this dropout regularizer, discussed its relationship with other regularizers and training algorithms, and evaluated it experimentally. Baldi and Sadowski (2014) exposed properties of dropout when viewed as an ensemble method (see also Bachman et al., 2014). Van Erven et al. (2014) showed that applying dropout for online learning in the experts setting leads to algorithms that adapt to important properties of the input without requiring doubling or other parameter-tuning techniques, and Abernethy et al. (2014) analyzed a class of methods including dropout by viewing these methods as smoothers. The impact of dropout on generalization (roughly, how much dropout restricts the search space of the learner, or, from a bias-variance point of view, its impact on variance) was studied by Wan et al. (2013) and Wager et al. (2014). The latter paper considers a variant of dropout compatible with a Poisson source, and shows that under some assumptions this dropout variant converges more quickly to its infinite sample limit than non-dropout training, and that the Bayes-optimal predictions are preserved under the modified dropout distribution. Our results complement theirs by focusing on the effect of the original dropout on the algorithm's bias.

Section 2 defines our notation and characterizes when the dropout criterion has a unique minimizer. Section 3 presents many additional properties of the dropout regularizer. Section 4 formally defines when two distributions separate two algorithms or regularizers. Sections 5 and 6 give sources over $\mathbf{R}^2$ that separate dropout and L_2; these exploit the preference of dropout for hypotheses that concentrate weight on a single feature. Section 7 provides plots demonstrating that the same distributions separate dropout from L_1 regularization. Section 8 gives a definition of co-adaptation and shows (using plots) that distributions exploiting dropout's bias against co-adapted weights can also be used to separate dropout from L_2 and L_1 regularization. Sections 9 and 10 give additional separation results using distributions with many features.

2. Preliminaries

We use $\mathbf{w}^*$ for the optimizer of the dropout criterion, q for the probability that a feature is dropped out, and $p = 1 - q$ for the probability that a feature is kept throughout the paper.

As in the introduction, if $X \subseteq \mathbf{R}^n$ and P is a joint distribution over $X \times \{-1, 1\}$, define

$$\mathbf{w}^*(P, q) \stackrel{\text{def}}{=} \text{argmin}_{\mathbf{w}} \mathbf{E}_{(\mathbf{x},y)\sim P, \boldsymbol{\nu}}(\ell(y\mathbf{w} \cdot (\mathbf{x} + \boldsymbol{\nu}))) \tag{3}$$

where $\nu_i = b_i x_i$ for $b_1, ..., b_n$ sampled independently at random from $\{-1, 1/p - 1\}$ with $\mathbf{Pr}(b_i = 1/p - 1) = p = 1 - q$, and $\ell(z)$ is the logistic loss function:

$$\ell(z) = \ln(1 + \exp(-z)).$$

For some analyses, an alternative representation of $\mathbf{w}^*(P, q)$ will be easier to work with. Let $r_1, ..., r_n$ be sampled randomly from $\{0, 1\}$, independently of $(\mathbf{x}, y)$ and one another, with $\mathbf{Pr}(r_i = 1) = p$. Defining $\mathbf{r} \odot \mathbf{x} = (x_1 r_1, ..., x_n r_n)$, we have the equivalent definition

$$\mathbf{w}^*(P, q) = p \, \text{argmin}_{\mathbf{w}} \mathbf{E}_{(\mathbf{x},y)\sim P, \mathbf{r}}(\ell(y\mathbf{w} \cdot (\mathbf{r} \odot \mathbf{x}))). \tag{4}$$

To see that they are equivalent, note that

$$\begin{aligned} \mathbf{E}(\ell(y\mathbf{w} \cdot (\mathbf{x} + \boldsymbol{\nu}))) &= \mathbf{E}\left(\ell\left(y\mathbf{w} \cdot \left(\frac{\mathbf{r} \odot \mathbf{x}}{p}\right)\right)\right) \\ &= \mathbf{E}(\ell(y(\mathbf{w}/p) \cdot (\mathbf{r} \odot \mathbf{x}))). \end{aligned}$$

Although this paper focuses on the logistic loss, the above definitions can be used for any loss function $\ell()$. Since the dropout criterion is an expectation of $\ell()$, we have the following obvious consequence.

Proposition 1 *If loss $\ell(\cdot)$ is convex, then the dropout criterion is also a convex function of $\mathbf{w}$.*

The remainder of the paper focuses on the logistic loss, $\ell(y\mathbf{w} \cdot \mathbf{x}) = \ln(1 + \exp(-y\mathbf{w} \cdot \mathbf{x}))$. We use $\mathbf{v}$ for the optimizer of the L_2 regularized criterion:

$$\mathbf{v}(P, \lambda) \stackrel{\text{def}}{=} \text{argmin}_{\mathbf{w}} \mathbf{E}_{(\mathbf{x},y)\sim P}(\ell(y\mathbf{w} \cdot \mathbf{x})) + \frac{\lambda}{2}||\mathbf{w}||^2. \tag{5}$$

It is not hard to see that the $\frac{\lambda}{2}||\mathbf{w}||^2$ term implies that $\mathbf{v}(P, \lambda)$ is always well-defined. On the other hand, $\mathbf{w}^*(P, q)$ is *not* always well-defined, as can be seen by considering any distribution concentrated on a single example. This motivates the following definition.

Definition 2 *Let P be a joint distribution with support contained in $\mathbf{R}^n \times \{-1, 1\}$. A feature i is* perfect modulo ties *for P if either $yx_i \geq 0$ for all $\mathbf{x}$ in the support of P, or $yx_i \leq 0$ for all $\mathbf{x}$ in the support of P.*

Put another way, i is perfect modulo ties if there is a linear classifier that only pays attention to feature i and is perfect on the part of P where x_i is nonzero.

Proposition 3 *For all finite domains $X \subseteq \mathbf{R}^n$, all distributions P with support in X, and all $q \in (0, 1)$, we have that $\mathbf{E}_{(\mathbf{x},y)\sim P, \mathbf{r}}(\ell(y\mathbf{w} \cdot (\mathbf{r} \odot \mathbf{x})))$ has a unique minimum in $\mathbf{R}^n$ if and only if no feature is perfect modulo ties for P.*

$\mathbf{x} = (x_1, \ldots, x_n)$	feature vector in $\mathbf{R}^n$
y	label in $\{-1, +1\}$
$\mathbf{w} = (w_1, \ldots, w_n)$	weight vector in $\mathbf{R}^n$
$\ell(y\mathbf{w} \cdot \mathbf{x})$	loss function, generally the logistic loss: $\ln(1 + \exp(-y\mathbf{w} \cdot \mathbf{x}))$
P, Q	source distributions over $(\mathbf{x}, y)$ pairs, varies by section
D	marginal distribution over $\mathbf{x}$
q	feature dropout probability in $(0, 1)$
$p = 1 - q$	probability of keeping a feature
λ	L_2 regularization parameter
$\boldsymbol{\nu} = (\nu_1, \ldots, \nu_n)$	additive dropout noise, $\nu_i \in \{-x_i, x_i/p - x_i\}$
$\mathbf{r} = (r_1, \ldots, r_n)$	multiplicative dropout noise, $r_i \in \{0, 1\}$
$\odot$	component-wise product: $\mathbf{r} \odot \mathbf{x} = (r_1 x_1, \ldots, r_n x_n)$
$\mathbf{w}^*(P, q)$ and $\mathbf{w}^*$	minimizer of dropout criterion: $\mathbf{E}(\ell(y\ \mathbf{w} \cdot (\mathbf{x} + \boldsymbol{\nu})))$
$\mathbf{w}^{\circledast} = \mathbf{w}^*/p$	minimizer of expected loss $\mathbf{E}(\ell(y\ \mathbf{w} \cdot (\mathbf{r} \odot \mathbf{x})))$
$\mathbf{v}(P, \lambda)$ and $\mathbf{v}$	minimizer of L_2-regularized loss
$\mathbf{reg}_{D,q}(\mathbf{w})$	regularization due to dropout
J, K	criteria to be optimized, varies by sub-section
$g(\mathbf{w})$, $\mathbf{g}$	gradients of the current criterion
$\mathrm{er}_P(\mathbf{w})$	0-1 classification generalization error of $\mathrm{sign}(\mathbf{w} \cdot x)$

Table 1: Summary of notation used throughout the paper.

Proof: Assume for contradiction that feature i is perfect modulo ties for P and some $\mathbf{w}^{\circledast}$ is the unique minimizer of $\mathbf{E}_{(\mathbf{x},y)\sim P,\mathbf{r}}(\ell(y\mathbf{w} \cdot (\mathbf{r} \odot \mathbf{x})))$. Assume w.l.o.g. that $yx_i \geq 0$ for all $\mathbf{x}$ in the support of P (the case where $yx_i \leq 0$ is analogous). Increasing $w_i^{\circledast}$ keeps the loss unchanged on examples where $x_i = 0$ and decreases the loss on the other examples in the support of P, contradicting the assumption that $\mathbf{w}^{\circledast}$ was a unique minimizer of the expected loss.

Now, suppose then each feature i has both examples where $yx_i > 0$ and examples where $yx_i < 0$ in the support of P. Since the support of P is finite, there is a positive lower bound on the probability of any example in the support. With probability $p(1-p)^{n-1}$, component r_i of random vector $\mathbf{r}$ is non-zero and the remaining $n-1$ components are all zero. Therefore as w_i increases without bound in the positive or negative direction, $\mathbf{E}_{(\mathbf{x},y)\sim P,\mathbf{r}}(\ell(y\mathbf{w}\cdot(\mathbf{r}\odot\mathbf{x})))$ also increases without bound. Since $\mathbf{E}_{(\mathbf{x},y)\sim P,\mathbf{r}}(\ell(y\mathbf{0}\cdot(\mathbf{r}\odot\mathbf{x}))) = \ln 2$, there is a value M depending only on distribution P and the dropout probability such that minimizing $\mathbf{E}_{(\mathbf{x},y)\sim P,\mathbf{r}}(\ell(y\mathbf{w} \cdot (\mathbf{r} \odot \mathbf{x})))$ over $\mathbf{w} \in [-M, M]^n$ is equivalent to minimizing $\mathbf{E}_{(\mathbf{x},y)\sim P,\mathbf{r}}(\ell(y\mathbf{w}\cdot(\mathbf{r}\odot\mathbf{x})))$ over $\mathbf{R}^n$. Since $\mathbf{Pr}_{(\mathbf{x},y)}(x_i = 0) \neq 1$ for all i, $\{\mathbf{r}\odot\mathbf{x} : \mathbf{r} \in \{0,1\}^n, \mathbf{x} \in X\}$ has full rank and therefore $\mathbf{E}_{(\mathbf{x},y)\sim P,\mathbf{r}}(\ell(y\mathbf{w} \cdot (\mathbf{r} \odot \mathbf{x})))$ is strictly convex. Since a strictly convex function defined on a compact set has a unique minimum, $\mathbf{E}_{(\mathbf{x},y)\sim P,\mathbf{r}}(\ell(y\mathbf{w} \cdot (\mathbf{r} \odot \mathbf{x})))$ has a unique minimum on $[-M, M]^n$, and therefore on $\mathbf{R}^n$. ∎

See Table 1 for a summary of the notation used in the paper.

3. Properties of the Dropout Regularizer

We start by rederiving the regularization function corresponding to dropout training previously presented by Wager et al. (2013), specialized to our context and using our notation. The first step is to write $\ell(y\mathbf{w}\cdot\mathbf{x})$ in an alternative way that exposes some symmetries:

$$\begin{aligned}\ell(y\mathbf{w}\cdot\mathbf{x}) &= \ln(1+\exp(-y\mathbf{w}\cdot\mathbf{x}))\\ &= \ln\left(\frac{\exp(y(\mathbf{w}\cdot\mathbf{x})/2)+\exp(-y(\mathbf{w}\cdot\mathbf{x})/2)}{\exp(y(\mathbf{w}\cdot\mathbf{x})/2)}\right)\\ &= \ln\left(\frac{\exp((\mathbf{w}\cdot\mathbf{x})/2)+\exp(-(\mathbf{w}\cdot\mathbf{x})/2)}{\exp(y(\mathbf{w}\cdot\mathbf{x})/2)}\right). \qquad (6)\end{aligned}$$

This then implies

$$\begin{aligned}&\mathbf{reg}_{D,q}(\mathbf{w})\\ &= \mathbf{E}(\ell(y\mathbf{w}\cdot(\mathbf{x}+\boldsymbol{\nu}))) - \mathbf{E}(\ell(y\mathbf{w}\cdot\mathbf{x}))\\ &= \mathbf{E}\left(\ln\left(\frac{\exp((\mathbf{w}\cdot(\mathbf{x}+\boldsymbol{\nu}))/2)+\exp(-(\mathbf{w}\cdot(\mathbf{x}+\boldsymbol{\nu}))/2)}{\exp(y(\mathbf{w}\cdot(\mathbf{x}+\boldsymbol{\nu}))/2)}\times\frac{\exp(y(\mathbf{w}\cdot\mathbf{x})/2)}{\exp((\mathbf{w}\cdot\mathbf{x})/2)+\exp(-(\mathbf{w}\cdot\mathbf{x})/2)}\right)\right)\\ &= \mathbf{E}\Big(\ln\left(\frac{\exp((\mathbf{w}\cdot(\mathbf{x}+\boldsymbol{\nu}))/2)+\exp(-(\mathbf{w}\cdot(\mathbf{x}+\boldsymbol{\nu}))/2)}{\exp((\mathbf{w}\cdot\mathbf{x})/2)+\exp(-(\mathbf{w}\cdot\mathbf{x})/2)}\right)-y(\mathbf{w}\cdot\boldsymbol{\nu})/2\Big).\end{aligned}$$

Since $\mathbf{E}(\boldsymbol{\nu})=\mathbf{0}$, we get the following.

Proposition 4 *(Wager et al., 2013)*

$$\mathbf{reg}_{D,q}(\mathbf{w}) = \mathbf{E}\left(\ln\left(\frac{\exp(\mathbf{w}\cdot(\mathbf{x}+\boldsymbol{\nu})/2)+\exp(-\mathbf{w}\cdot(\mathbf{x}+\boldsymbol{\nu})/2)}{\exp((\mathbf{w}\cdot\mathbf{x})/2)+\exp(-(\mathbf{w}\cdot\mathbf{x})/2)}\right)\right). \qquad (7)$$

Using a Taylor expansion, Wager et al. (2013) arrived at the following approximation:

$$\frac{q}{2(1-q)}\sum_i w_i^2\mathbf{E}_{\mathbf{x}}\left(\frac{x_i^2}{(1+\exp(-\frac{\mathbf{w}\cdot\mathbf{x}}{2}))(1+\exp(\frac{\mathbf{w}\cdot\mathbf{x}}{2}))}\right). \qquad (8)$$

This approximation suggests two properties: the strength of the regularization penalty decreases exponentially in the prediction confidence $|\mathbf{w}\cdot\mathbf{x}|$, and that the regularization penalty goes to infinity as the dropout probability q goes to 1. However, $\mathbf{w}\cdot\boldsymbol{\nu}$ can be quite large, making a second-order Taylor expansion inaccurate.[1] In fact, the analysis in this section suggests that the regularization penalty does not decrease with the confidence and that the regularization penalty increases linearly with $q=1-p$ (Figure 1, Theorem 8, Proposition 9).

The following propositions show that $\mathbf{reg}_{D,q}(\mathbf{w})$ satisfies at least some of the intuitive properties of a regularizer.

Proposition 5 $\mathbf{reg}_{D,q}(\mathbf{0})=0$.

Proposition 6 *(Wager et al., 2013) The contribution of each* $\mathbf{x}$ *to the dropout regularization penalty (7) is non-negative: for all* $\mathbf{x}$*,*

$$\mathbf{E}_{\boldsymbol{\nu}}\left(\ln\left(\frac{\exp((\mathbf{w}\cdot(\mathbf{x}+\boldsymbol{\nu}))/2)+\exp(-(\mathbf{w}\cdot(\mathbf{x}+\boldsymbol{\nu}))/2)}{\exp((\mathbf{w}\cdot\mathbf{x})/2)+\exp(-(\mathbf{w}\cdot\mathbf{x})/2)}\right)\right)\geq 0.$$

1. Wager et al. (2013) experimentally evaluated the accuracy of a related approximation in the case that, instead of using dropout, $\boldsymbol{\nu}$ was distributed according to a zero-mean Gaussian.

Proof: The proposition follows from Jensen's Inequality. ∎

The $\mathbf{w}^*(P,q)$ vector learned by dropout training minimizes

$$\mathbf{E}_{(\mathbf{x},y)\sim P}(\ell(y\mathbf{w}\cdot\mathbf{x})) + \mathbf{reg}_{D,q}(\mathbf{w}).$$

However, the $\mathbf{0}$ vector has $\ell(y\mathbf{0}\cdot\mathbf{x}) = \ln(2)$ and $\mathbf{reg}_{D,q}(\mathbf{0}) = 0$, implying:

Proposition 7 $\mathbf{reg}_{D,q}(\mathbf{w}^*) \le \ln(2)$.

Thus any regularization penalty greater than $\ln(2)$ is effectively equivalent to a regularization penalty of ∞.

We now present new results based on analyzing the exact $\mathbf{reg}_{D,q}(\mathbf{w})$. The next properties show that the dropout regularizer is emphatically *not* like other convex or norm-based regularization penalties in that the dropout regularization penalty always remains bounded when a single component of the weight vector goes to infinity (see also Figure 1).

Theorem 8 *For all dropout probabilities $1-p \in (0,1)$, all n, all marginal distributions D over n-feature vectors, and all indices $1 \le i \le n$,*

$$\sup_{w_i} \mathbf{reg}_{D,q}(\underbrace{0,\ldots,0}_{i-1}, w_i, \underbrace{0,\ldots,0}_{n-i}) \le \mathbf{Pr}_D(x_i \neq 0)(1-p)\ln(2) \;<\; \ln 2.$$

Proof: Fix arbitrary n, p, i, and D. We have

$$\begin{aligned}&\mathbf{reg}_{D,q}(\underbrace{0,\ldots,0}_{i-1}, w_i, \underbrace{0,\ldots,0}_{n-i})\\ &= \mathbf{E}_{\mathbf{x},\boldsymbol{\nu}}\left(\ln\left(\frac{\exp(-w_i(x_i+\nu_i)/2)+\exp(w_i(x_i+\nu_i)/2)}{\exp(-w_ix_i/2)+\exp(w_ix_i/2)}\right)\right).\end{aligned}$$

Fix an arbitrary $\mathbf{x}$ in the support of D and examine the expectation over $\boldsymbol{\nu}$ for that $\mathbf{x}$. Recall that $x_i+\nu_i$ is 0 with probability $1-p$ and is x_i/p with probability p, and we will use the substitution $z = |w_ix_i|/2$.

$$\mathbf{E}_{\boldsymbol{\nu}}\left(\ln\left(\frac{\exp(\frac{-w_i(x_i+\nu_i)}{2})+\exp(\frac{w_i(x_i+\nu_i)}{2})}{\exp(\frac{-w_ix_i}{2})+\exp(\frac{w_ix_i}{2})}\right)\right) \tag{9}$$

$$= (1-p)\ln(2) + p\ln\left(\exp(\frac{z}{p})+\exp(\frac{-z}{p})\right) - \ln\left(\exp(z)+\exp(-z)\right). \tag{10}$$

We now consider cases based on whether or not z is 0. When $z=0$ (so either w_i or x_i is 0) then (10) is also 0.

If $z \neq 0$ then consider the derivative of (10) w.r.t. z, which is

$$\frac{\exp(z/p)-\exp(-z/p)}{\exp(z/p)+\exp(-z/p)} - \frac{\exp(z)-\exp(-z)}{\exp(z)+\exp(-z)}.$$

This derivative is positive since $z>0$ and $0<p<1$. Therefore (10) is bounded by its limit as $z\to\infty$, which is $(1-p)\ln(2)$, in this case.

Since (9) is 0 when $x_i=0$ and is bounded by $(1-p)\ln(2)$ otherwise, the expectation over $\mathbf{x}$ of (9) is bounded $\mathbf{Pr}_D(x_i\neq 0)(1-p)\ln(2)$, completing the proof. ∎

Since line (10) is derived using a chain of equalities, the same proof ideas can be used to show that Theorem 8 is tight.

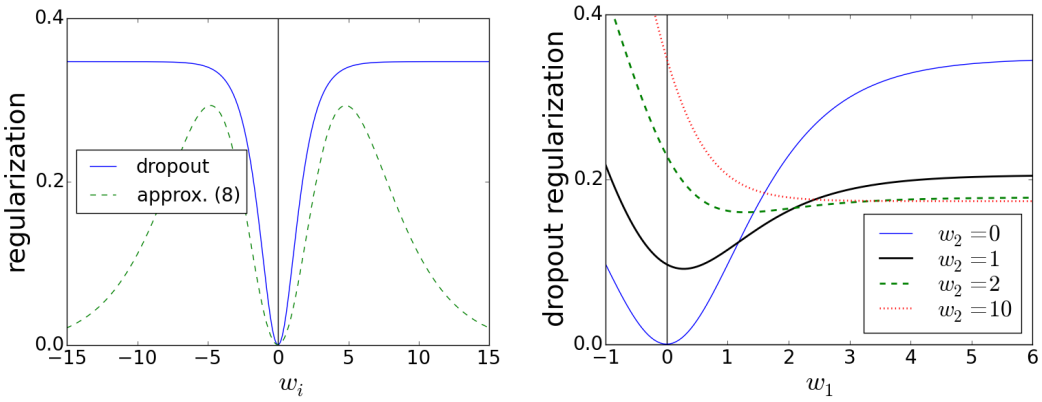


Figure 1: The $p = 1/2$ dropout regularization for $\mathbf{x} = (1, 1)$ as a function of w_i when the other weights are 0 together with its approximation (8) (left) and as a function of w_1 for different values of the second weight (right).

Proposition 9 *Under the conditions of Theorem 8,*

$$\lim_{w_i \to \infty} \mathbf{reg}_{D,q}(\underbrace{0, \ldots, 0}_{i-1}, w_i, \underbrace{0, \ldots, 0}_{n-i}) = \mathbf{Pr}_D(x_i \neq 0)(1-p)\ln(2).$$

Note that this bound on the regularization penalty depends neither on the range nor expectation of x_i. In particular, it has a far different character than the approximation of Equation (8).

In Theorem 8 the other weights are fixed at 0 as w_i goes to infinity. An additional assumption implies that the regularization penalty remains bounded even when the other components are non-zero. Let $\mathbf{w}$ be a weight vector such that for all $\mathbf{x}$ in the support of D and dropout noise vectors $\boldsymbol{\nu}$ we have $|\sum_{j \neq i} w_j(x_j + \nu_j)| \leq M$ for some bound M (this implies that $|\sum_{j \neq i} w_j x_j| \leq M$ also). Then

$$\begin{aligned}
\mathbf{reg}_{D,q}(\mathbf{w}) &= \mathbf{E}_{\mathbf{x},\boldsymbol{\nu}} \left(\left(\frac{\exp(\frac{\mathbf{w}\cdot(\mathbf{x}+\boldsymbol{\nu})}{2}) + \exp(-\frac{\mathbf{w}\cdot(\mathbf{x}+\boldsymbol{\nu})}{2})}{\exp(\frac{\mathbf{w}\cdot\mathbf{x}}{2}) + \exp(-\frac{\mathbf{w}\cdot\mathbf{x}}{2})} \right) \right) \\
&\leq \mathbf{E}_{x_i,\nu_i} \left(\log \left(\frac{\exp(\frac{M - w_i(x_i+\nu_i)}{2}) + \exp(\frac{M + w_i(x_i+\nu_i)}{2})}{\exp(-\frac{M - w_i x_i}{2}) + \exp(-\frac{M + w_i x_i}{2})} \right) \right) \\
&\leq M + \mathbf{E}_{x_i,\nu_i} \left(\log \left(\frac{\exp(-\frac{w_i x_i + \nu_i}{2}) + \exp(\frac{w_i(x_i+\nu_i)}{2})}{\exp(\frac{-w_i x_i}{2}) + \exp(\frac{w_i x_i}{2})} \right) \right). \qquad (11)
\end{aligned}$$

Using (11) instead of the first line in Theorem 8's proof gives the following.

Proposition 10 *Under the conditions of Theorem 8, if the weight vector* $\mathbf{w}$ *has the property that* $|\sum_{j \neq i} w_j(x_j + \nu_j)| \leq M$ *for each* $\mathbf{x}$ *in the support of* D *and all of its corresponding dropout noise vectors* $\boldsymbol{\nu}$ *then*

$$\sup_{\omega} \mathbf{reg}_{D,q}(w_1, w_2, \ldots, w_{i-1}, \omega, w_{i+1}, \ldots, w_n) \leq M + \mathbf{Pr}_D(x_i \neq 0)(1-p)\ln(2).$$

Proposition 10 shows that the regularization penalty starting from a non-zero initial weight vector remains bounded as any one of its components goes to infinity. On the other hand, unless M is small, the bound will be larger than the dropout criterion for the zero vector. This is a natural consequence as the starting weight vector $\mathbf{w}$ could already have a large regularization penalty.

The derivative of (10) in the proof of Theorem 8 implies that the dropout regularization penalty is monotonic in $|w_i|$ when the other weights are zero. Surprisingly, this is does *not* hold in general. The dropout regularization penalty due to a single example (as in Proposition 6) can be written as

$$\mathbf{E}_{\boldsymbol{\nu}}\left(\ln\left(\exp(\tfrac{\mathbf{w}\cdot(\mathbf{x}+\boldsymbol{\nu})}{2})+\exp(\tfrac{-\mathbf{w}\cdot(\mathbf{x}+\boldsymbol{\nu})}{2})\right)\right)-\ln\left(\exp(\tfrac{\mathbf{w}\cdot\mathbf{x}}{2})+\exp(\tfrac{-\mathbf{w}\cdot\mathbf{x}}{2})\right).$$

Therefore if increasing a weight makes the second logarithm increase faster than the expectation of the first, then the regularization penalty decreases even as the weight increases. This happens when the $w_i x_i$ products tend to have the same sign. The regularization penalty as a function of w_1 for the single example $\mathbf{x}=(1,1)$, $p=1/2$, and w_2 set to various values is plotted in Figure 1.[2] This gives us the following.

Proposition 11 *Unlike p-norm regularizers, the dropout regularization penalty* $\mathbf{reg}_{D,q}(\mathbf{w})$ *is not always monotonic in the individual weights.*

In fact, the dropout regularization penalty can decrease as weights move up from 0.

Proposition 12 *Fix* $p=1/2$, $w_2>0$, *and an arbitrary* $\mathbf{x}\in(0,\infty)^2$. *Let* D *be the distribution concentrated on* $\mathbf{x}$. *Then* $\mathbf{reg}_{D,q}(w_1,w_2)$ *locally decreases as* w_1 *increases from* 0.

Proposition 12 is proved in Appendix A.

We now turn to the dropout regularization's behavior when two weights vary together. If any features are always zero then their weights can go to $\pm\infty$ without affecting either the predictions or $\mathbf{reg}_{D,q}(\mathbf{w})$. Two linearly dependent features might as well be one feature. After ruling out degeneracies like these, we arrive at the following theorem, which is proved in Appendix B.

Theorem 13 *Fix an arbitrary distribution* D *with support in* $\mathbf{R}^2$, *weight vector* $\mathbf{w}\in\mathbf{R}^2$, *and non-dropout probability* p. *If there is an* $\mathbf{x}$ *with positive probability under* D *such that* w_1x_1 *and* w_2x_2 *are both non-zero and have different signs, then the regularization penalty* $\mathbf{reg}_{D,q}(\omega\mathbf{w})$ *goes to infinity as* ω *goes to* $\pm\infty$.

The theorem can be straightforwardly generalized to the case $n>2$; except in degenerate cases, sending two weights to infinity together will lead to a regularization penalty approaching infinity.

Theorem 13 immediately leads to the following corollary.

2. Setting $\mathbf{x}=(1,1)$ is in some sense without loss of generality as the prediction and dropout regularization values for any $\mathbf{w}$, $\mathbf{x}$ pair are identical to the values for $\tilde{\mathbf{w}}$, $\mathbf{1}$ when each $\tilde{w}_i=w_ix_i$.

Corollary 14 *For a distribution D with support in $\mathbf{R}^2$, if there is an $\mathbf{x}$ with positive probability under D such that $x_1 \neq 0$ and $x_2 \neq 0$, then there is a $\mathbf{w}$ such that for any $q \in (0,1)$, the regularization penalty $\mathbf{reg}_{D,q}(\omega\mathbf{w})$ goes to infinity with ω.*

For any $\mathbf{w} \in \mathbf{R}^2$ with both components nonzero, there is a distribution D over $\mathbf{R}^2$ with bounded support such that the regularization penalty $\mathbf{reg}_{D,q}(\omega\mathbf{w})$ goes to infinity with ω.

Together Theorems 8 and 13 demonstrate that $\mathbf{reg}_{D,q}(\mathbf{w})$ is *not* convex (see also Figure 1). In fact, $\mathbf{reg}_{D,q}(\mathbf{w})$ cannot be approximated to within any factor by a convex function, even if a dependence on n and p is allowed. For example, Theorem 8 shows that, for all D with bounded support, both $\mathbf{reg}_{D,q}(0,\omega)$ and $\mathbf{reg}_{D,q}(\omega,0)$ remain bounded as ω goes to infinity, whereas Theorem 13 shows that there is such a D such that $\mathbf{reg}_{D,q}(\omega/2,\omega/2)$ is unbounded as ω goes to infinity.

Theorem 13 relies on the $w_i x_i$ products having different signs. The following shows that $\mathbf{reg}_{D,q}(\mathbf{w})$ does remain bounded when multiple components of $\mathbf{w}$ go to infinity if the corresponding features are compatible in the sense that the signs of $w_i x_i$ are always in alignment.

Theorem 15 *Let $\mathbf{w}$ be a weight vector and D be a discrete distribution such that $w_i x_i \geq 0$ for each index i and all $\mathbf{x}$ in the support of D. The limit of $\mathbf{reg}_{D,q}(\omega\mathbf{w})$ as ω goes to infinity is bounded by $\ln(2)(1-p)\mathbf{P}_{\mathbf{x}\sim D}(\mathbf{w}\cdot\mathbf{x} \neq 0)$.*

The proof of Theorem 15 (which is Appendix C) easily generalizes to alternative conditions where $\omega \to -\infty$ and/or $w_i x_i \leq 0$ for each $i \leq k$ and $\mathbf{x}$ in the support of D.

Taken together Theorems 15 and 13 give an almost complete characterization of when multiple weights can go to infinity while maintaining a finite dropout regularization penalty.

3.1 Discussion

The bounds in the preceding theorems and propositions suggest several properties of the dropout regularizer. First, the $1-p$ factors indicate that the strength of regularization grows linearly with dropout probability $q = 1-p$. Second, the $\mathbf{P}_{\mathbf{x}\sim D}(x_i \neq 0)$ factors in several of the bounds suggest that weights for rare features are encouraged by being penalized less strongly than weights for frequent features. This preference for rare features is sometimes seen in algorithms like the Second-Order Perceptron (Cesa-Bianchi et al., 2002) and AdaGrad (Duchi et al., 2011). Wager et al. (2013) discussed the relationship between dropout and these algorithms, based on approximation (8). Empirical results indicate that dropout performs well in domains like document classification where rare features can have high discriminative value (Wang and Manning, 2013). The theorems of this section suggest that the exact dropout regularizer minimally penalizes the use of rare features. Finally, Theorem 13 suggests that dropout limits co-adaptation by strongly penalizing large weights if the $w_i x_i$ products often have different signs. On the other hand, if the $w_i x_i$ products usually have the same sign, then Proposition 12 indicates that dropout encourages increasing the smaller weights to help share the prediction responsibility. This intuition is reinforced by Figure 1, where the dropout penalty for two large weights is much less then a single large weight when the features are highly correlated.

4. A Definition of Separation

Now we turn to illustrating the inductive bias of dropout by contrasting it with L_2 regularization. For this, we will use a definition of separation between pairs of regularizers.

Each regularizer has a regularization parameter that governs how strongly it regularizes. If we want to describe qualitatively what is preferred by one regularizer over another, we need to control for the amount of regularization.

Let $\mathrm{er}_P(\mathbf{w}) = \mathbf{Pr}_{(\mathbf{x},y)\sim P}(\mathrm{sign}(\mathbf{w}\cdot\mathbf{x}) \neq y)$, and recall that $\mathbf{w}^*$ and $\mathbf{v}$ are the minimizers of the dropout and L_2-regularized criteria respectively.

Say that sources P and Q C-separate L_2 and dropout if there exist q and λ such that both $\frac{\mathrm{er}_P(\mathbf{w}^*(P,q))}{\mathrm{er}_P(\mathbf{v}(P,\lambda))} > C$ and $\frac{\mathrm{er}_Q(\mathbf{v}(Q,\lambda))}{\mathrm{er}_Q(\mathbf{w}^*(Q,q))} > C$. Say that indexed families $\mathcal{P} = \{P_\alpha\}$ and $\mathcal{Q} = \{Q_\alpha\}$ *strongly separate* L_2 and dropout if pairs of distributions in the family C-separate them for arbitrarily large C. We provide strong separations, using both $n = 2$ and larger n.

5. A Source Preferred by L_2

Consider the joint distribution P_5 defined as follows[3]:

x_1	x_2	y	$\mathbf{Pr}(\mathbf{x}, y)$
10	-1	1	1/3
1.1	-1	1	1/3
-1	1.1	1	1/3

(12)

This distribution has weight vectors that classify examples perfectly (the green shaded region in Figure 2). For this distribution, optimizing an L_2-regularized criterion leads to a perfect hypothesis[4], while the weight vectors optimizing the dropout criterion make prediction errors on one-third of the distribution.

The intuition behind this behavior for the distribution described in (12) is that weight vectors that are positive multiples of $(1, 1)$ classify all of the data correctly. However, with dropout regularization the $(10, -1)$ and $(1.1, -1)$ data points encourage the second weight to be negative when the first component is dropped out. This negative push on the second weight is strong enough to prevent the minimizer of the dropout-regularized criterion from correctly classifying the $(-1, 1.1)$ data point. Figure 2 illustrates the loss, dropout regularization, and dropout and L_2 criterion for this data source.[5]

3. Although several of our sources have all positive instances, that is not essential for the construction. The probability on each $(\mathbf{x}, y)$ example can be split evenly between the original $(\mathbf{x}, y)$ and its negatively-labeled counterpart $(-\mathbf{x}, -y)$. Note that for any $\mathbf{w}$, both $(\mathbf{x}, y)$ and its counterpart $(-\mathbf{x}, -y)$ make the same contribution to both the loss and dropout regularization. After splitting all of the examples, both labels will be equally represented in the distribution. Furthermore, with such paired examples, convexity implies that the weight on any non-dropped out bias input will be 0 when the criterion is minimized.
4. Having the labels of this distribution be consistent with a linear threshold function eases discussion, but is not essential. Adding a fourth inconsistent point with sufficiently small probability would preserve the property that the L_2-regularized criterion leads to a minimum error linear threshold hypothesis while the error of dropout's hypothesis is significantly larger.
5. The contours in this and the subsequent figures are not evenly spaced, but chosen to emphasize interesting aspects of the surfaces while minimizing clutter.

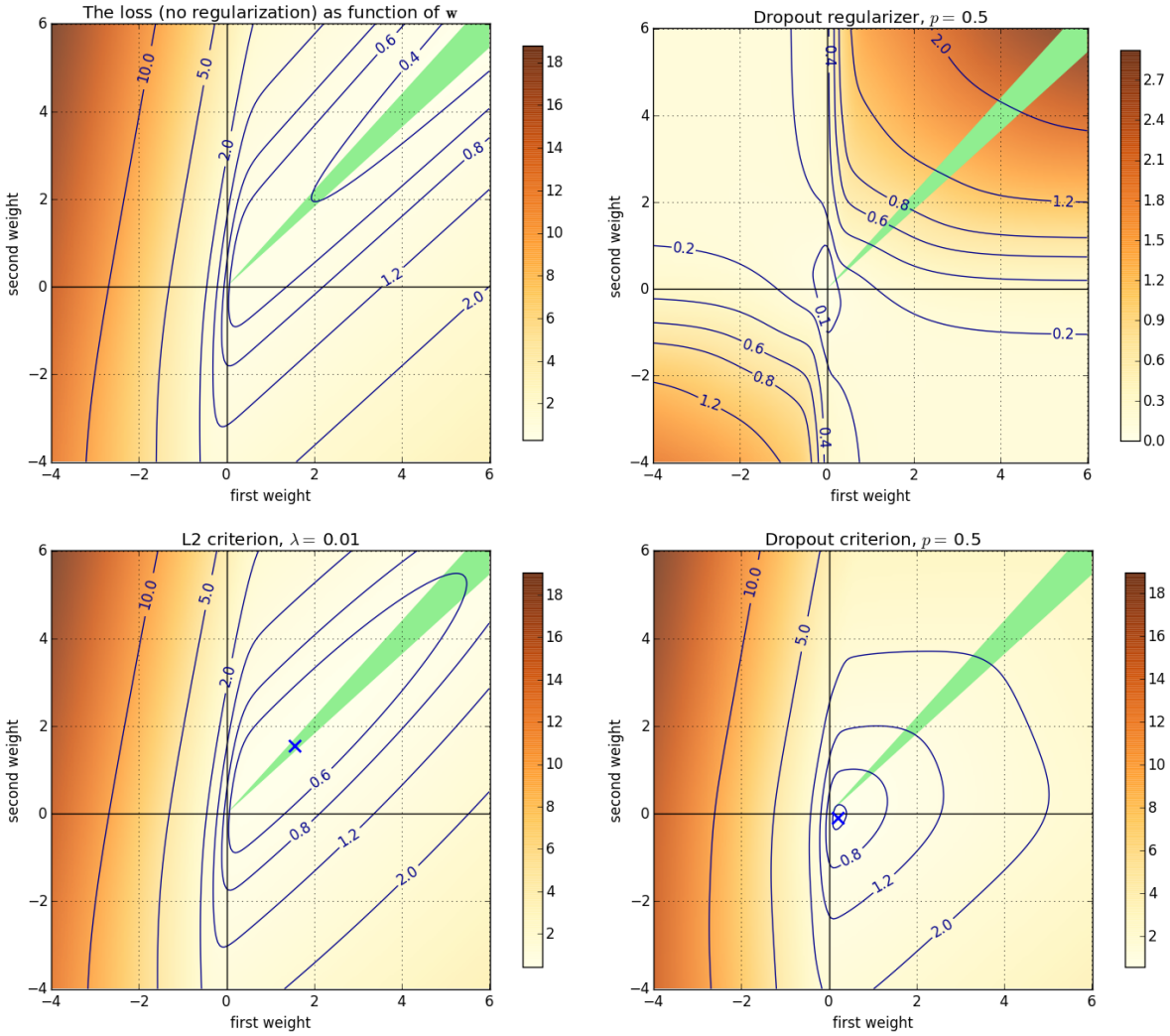


Figure 2: Using data favoring L_2 in (12). The expected loss is plotted in the upper-left, the dropout regularizer in the upper-right, the L_2 regularized criterion as in (5) in the lower-left and the dropout criterion as in (3) in the lower-right, all as functions of the weight vector. The Bayes-optimal weight vectors are in the green region, and "$\times$" marks show the optimizers of the criteria.

We first show that distribution P_5 of (12) is compatible with mild enough L_2 regularization. Recall that $\mathbf{v}(P_5, \lambda)$ is weight vector found by minimizing the L_2 regularized criterion (5).

Theorem 16 *If $0 < \lambda \leq 1/50$, then $\mathrm{er}_{P_5}(\mathbf{v}(P_5, \lambda)) = 0$ for the distribution P_5 defined in (12).*

In contrast, the $\mathbf{w}^*(P_5, q)$ minimizing the dropout criterion (3) has error rate at least 1/3.

Theorem 17 *If $q \geq 1/3$ then $\mathrm{er}_{P_5}(\mathbf{w}^*(P_5, q)) \geq 1/3$ for the distribution P_5 defined in (12).*

The proofs of Theorem 16 and 17 are in Appendices D and E.

6. A Source Preferred by Dropout

In this section, consider the joint distribution P_6 defined by

x_1	x_2	y	$\mathbf{Pr}(\mathbf{x}, y)$
1	0	1	3/7
$-1/1000$	1	1	3/7
1/10	-1	1	1/7

(13)

The intuition behind this distribution is that the $(1, 0)$ data point encourages a large weight on the first feature. This means that the negative pressure on the second weight due to the $(1/10, -1)$ data point is much smaller (especially given its lower probability) than the positive pressure on the second weight due to the $(-1/1000, 1)$ example. The L_2 regularized criterion emphasizes short vectors, and prevents the first weight from growing large enough (relative to the second weight) to correctly classify the $(1/10, -1)$ data point. On the other hand, the first feature is nearly perfect; it only has the wrong sign on the second example where it is $-\epsilon = -1/1000$. This means that, in light of Theorem 8 and Proposition 10, dropout will be much more willing to use a large weight for x_1, giving it an advantage for this source over L_2. The plots in Figure 3 illustrate this intuition.

Theorem 18 *If $1/100 \leq \lambda \leq 1$, then $\mathrm{er}_{P_6}(\mathbf{v}(P_6, \lambda)) \geq 1/7$ for the distribution P_6 defined in (13).*

In contrast, the minimizer of the dropout criterion is able to generalize perfectly.

Theorem 19 *If $q \leq 1/2$, then $\mathrm{er}_{P_6}(\mathbf{w}^*(P_6, q)) = 0$ for the distribution P_6 defined in (13).*

Theorems 18 and 19 are proved in Appendices F and G.

The results in this and the previous section show that the distributions defined in (12) and (13) strongly separate dropout and L_2 regularization. Theorem 19 shows that for distribution P analyzed in this section $\mathrm{er}_P(\mathbf{w}^*(P, q)) = 0$ for all $q \leq 1/2$ while Theorem 18 shows that for the same distribution $\mathrm{er}_P(\mathbf{v}(P, \lambda) \geq 1/7$ whenever $\lambda \geq 1/100$. In contrast, when Q is the distribution defined in the previous section, Theorem 16 shows $\mathrm{er}_Q(\mathbf{v}(Q, \lambda)) = 0$ whenever $\lambda \leq 1/50$. For this same distribution Q, Theorem 17 shows that $\mathrm{er}_Q(\mathbf{w}^*(Q, q)) \geq 1/3$ whenever $q \geq 1/3$.

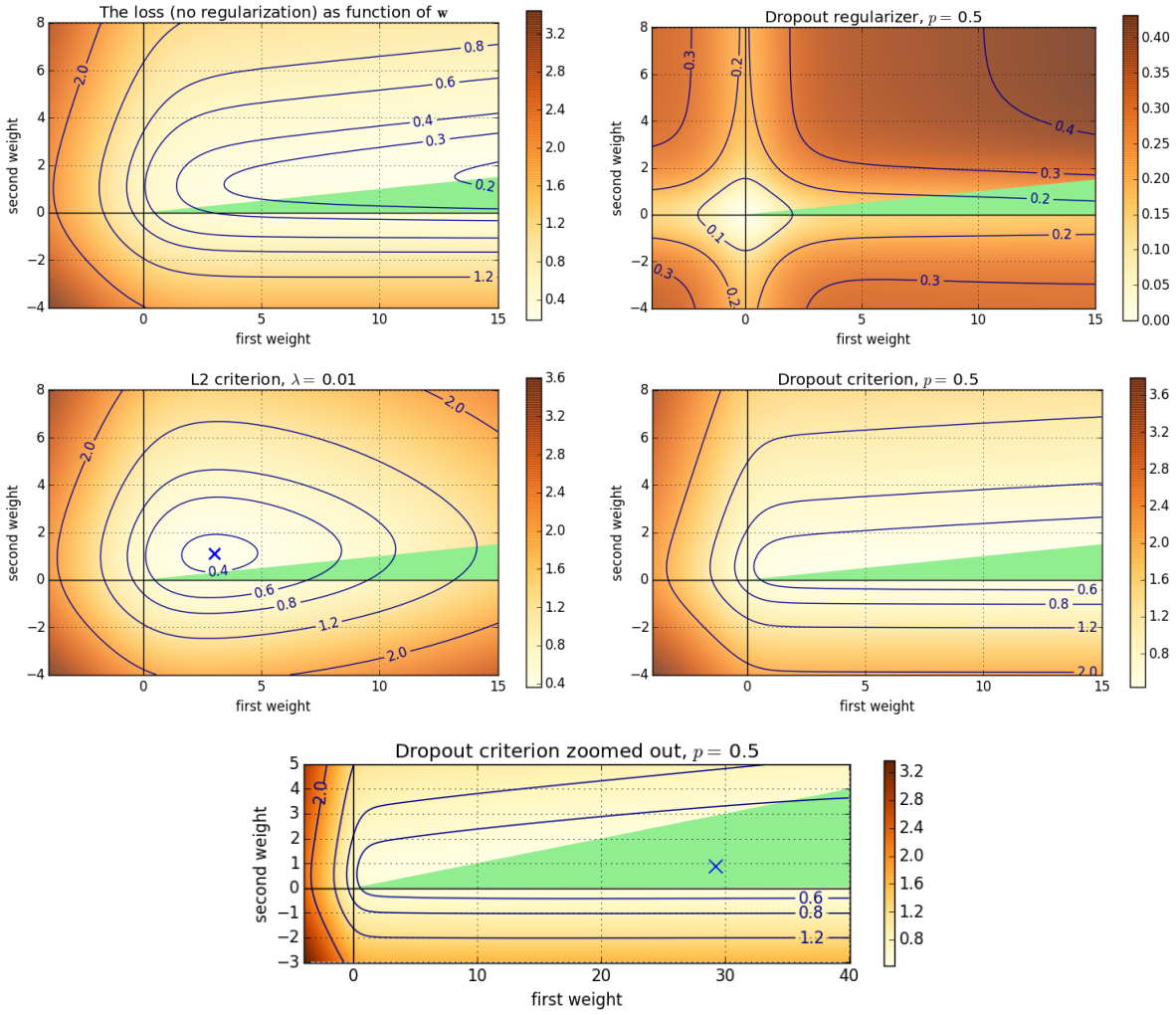


Figure 3: For the source from (13) favoring the dropout, the expected loss is plotted in the upper-left, the dropout regularizer in the upper-right, the expected loss plus L_2 regularization as in (5) in the lower-left and the dropout criterion as in (3) in the lower-right, all as functions of the weight vector. The Bayes-optimal weight vectors are in the green region, and "$\times$" marks show the optimizers of the criteria. Note that the minimizer of the dropout criterion lies outside the middle-right plot and is shown on the bottom plot (which has a different range and scale than the others.)

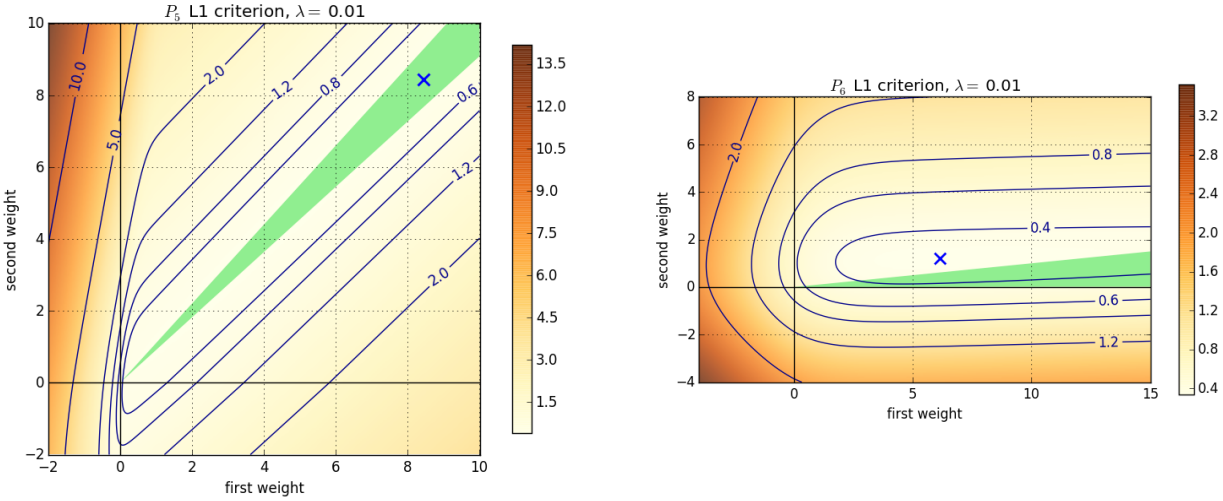


Figure 4: A plot of the L_1 criterion with $\lambda = 0.01$ for distributions P_5 defined in Section 5 (left) and P_6 defined in Section 6 (right). As before, the Bayes optimal classifiers are denoted by the region shaded in green and the minimizer of the criterion is denoted with an x.

7. L_1 Regularization

In this section, we show that the same P_5 and P_6 distributions that separate dropout from L_2 regularization also separate dropout from L_1 regularization: the algorithm the minimizes

$$\mathbf{E}_{(\mathbf{x},y)\sim P}(\ell(y\mathbf{w}\cdot\mathbf{x})) + \lambda||\mathbf{w}||_1. \tag{14}$$

As in Sections 5 and 6, we set $\lambda = 1/100$. Figure 4 plots the L_1 criterion (14) for the distributions P_5 defined in (12) and P_6 defined in (13). Like L_2 regularization, L_1 regularization produces a Bayes-optimal classifier on P_5, but not on P_6. Therefore the same argument shows that these distributions also strongly separate dropout and L_1 regularization.

8. Dropout and Co-adaptation

Hinton et al. (2012) and Srivastava et al. (2014) give evidence that dropout helps prevent the co-adaptation of units in neural networks, encouraging individual units to learn simpler functions of their inputs. In this section we provide a definition of co-adaptation and illustrate how dropout training can restrict the co-adaptation of weights.

We say that two weights w_i and w_j are co-adapted in a weight vector $\mathbf{w}$ if either alone increases the loss, but both together decrease the loss. More formally, let "$\mathbf{w} \setminus i$" denote vector $\mathbf{w}$ modified by replacing w_i with 0, and "$\mathbf{w} \setminus i, j$" denote the resulting vector when both w_i and w_j are replaced by 0. If all of:

1. The loss of $\mathbf{w}$ is less than the loss of $\mathbf{w} \setminus i, j$,
2. The loss of $\mathbf{w} \setminus i, j$ is less than the loss of $\mathbf{w} \setminus i$, and

3. The loss of $\mathbf{w} \setminus i, j$ is less than the loss of $\mathbf{w} \setminus j$,

then we say that weights w_i and w_j are *co-adapted* in $\mathbf{w}$.

For example, consider the case when features x_1 and x_2 tend to have the same sign, but x_1 is usually a little bigger than x_2 when the label is $+$, and x_2 tends to be larger when the label is $-$. Then the difference $x_1 - x_2$ usually has the same sign as the label, and making w_1 large and w_2 negative with a similar-magnitude is likely to decrease the loss. This is similar to constructing the new good feature $x_1 - x_2$ and giving it large weight. However, if neither feature x_1 nor feature x_2 is strongly correlated with the label, then using a large magnitude weight on just x_1 or just x_2 is likely to result in many badly misclassified examples, and greater loss than if w_1 and w_2 were both set to zero. (Note that similar co-adaptation situations arise when x_1 and x_2 have different signs, but their sum tends to have the same sign, or tends to have the opposite sign, as the label.)

Theorem 13 shows that the dropout regularization penalty goes to infinity as the opposite-signed weights given to x_1 and x_2 in the situation described. Furthermore, the dropout penalty for weight vector $\mathbf{w}$ includes terms for the loss of $\mathbf{w} \setminus i$ and $\mathbf{w} \setminus j$, so if these grow too large, then $\mathbf{w}$ cannot be the minimizer of the dropout criterion. This suggests that dropout training minimizes co-adaptation. The following example gives a more concrete illustration of this behavior.

Consider the joint distribution P_8 defined as follows:

x_1	x_2	y	$\mathbf{Pr}(\mathbf{x}, y)$
10	9	1	0.64
9	10	-1	0.32
1.0	-0.35	1	0.03
-0.35	1.0	-1	0.01

(15)

The loss and dropout regularization for P_8 are plotted in Figure 5. To obtain small loss, the hypothesis must give weights a similar large magnitude with w_1 positive while w_2 is negative. On the other hand, almost all of the probability is on the first two examples, and giving the weights different signs satisfies the conditions of Theorem 13 for them, and the dropout penalty quickly increases. The low probability points will also make the dropout regularization for weight vectors $\mathbf{w} = (a, a)$ go to infinity as a goes to infinity, but the small probabilities keeps the penalty small until a becomes very large (e.g. for $\mathbf{w} = (30, 30)$, the penalty is still less than 0.4). Omitting these points has a nearly indistinguishable effect on the first plots: their presence, as well as the different probabilities for the points, will be more important later, when we introduce the alternative labeling P_8'.

The "checkerboard" pattern of the regularization in Figure 5 shows that common patterns in the data can strongly shape the dropout regularizer, making it discriminate against certain directions. In Figure 6 we plot the L_1, L_2, and dropout regularized criteria for source P_8, illustrating that the dropout regularizer forces the weight vector away from the Bayes optimal region. In fact, the regularization is so strong that both weights are positive at the minimizer of the dropout criterion.

We can verify that the minimizing $\mathbf{v} \approx (2.8, -2.75)$ for the L_2 criterion exhibits co-adaptation. The loss of $\mathbf{v}$ is about 0.06, the loss of $\mathbf{v} \setminus 1 \approx 15$, the loss of $\mathbf{w} \setminus 2 \approx 8$, and the loss of $\mathbf{v} \setminus 1, 2 = \ln 2 \approx 0.69$. The co-adaptation is even more dramatic for the L_1 criterion.

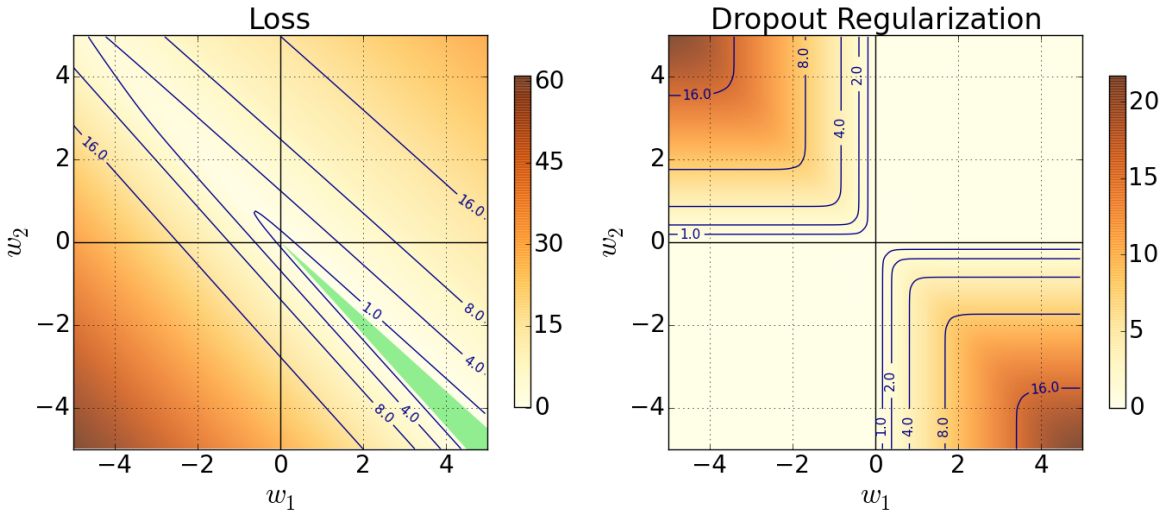


Figure 5: Plots of the loss and $p = 1/2$ dropout regularization for distribution P_8. Note that the regularization penalty increases quickly when the weights have opposite signs, but much more slowly when they have the same sign. In the loss plot, the green region indicates the Bayes optimal classifiers.

On the other hand, the weight vector $\mathbf{w}^* \approx (0.035, 0.014)$ minimizing the dropout criterion is not co-adapted. The losses of $\mathbf{w}^* \setminus 1$ and $\mathbf{w}^* \setminus 2$ are both greater than the loss of $\mathbf{w}^*$, but both are also *less* than the loss of $\mathbf{w}^* \setminus 1, 2$.

Although minimizing the dropout criterion fails to yield a Bayes optimal weight vector for P_8, the situation reverses when we consider the modified distribution P_8' with the same feature vectors and probabilities as P_8, but with all all labels set to 1. When all the labels are positive, the heavier points on the right pull the weight vector in that direction. If it is pulled far enough, then the (-0.3, 1) point will be misclassified.

Since the dropout regularization penalty depends only on the instance probabilities and not on the labels, P_8 and P_8' have the same regularization penalty function. The difference is that P_8' with its modified labels has low loss when both weights are large, a situation compatible with the dropout regularization. See Figure 7 for plots of the loss and various criteria for the modified P_8' source.

The plots in Figures 6 and 7 show that distributions P_8 and P_8' also strongly separate dropout from both L_2 and L_1 regularization. Since the two distributions have the same marginal distribution over feature vectors (and thus use the same dropout regularization penalty function), they provide vivid evidence of how dropout shapes the landscape, encouraging some directions while heavily penalizing others.

9. A High-Dimensional Source Preferred by L_2

In this section we exhibit a source where L_2 regularization leads to a perfect predictor while dropout regularization creates a predictor with a constant error rate.

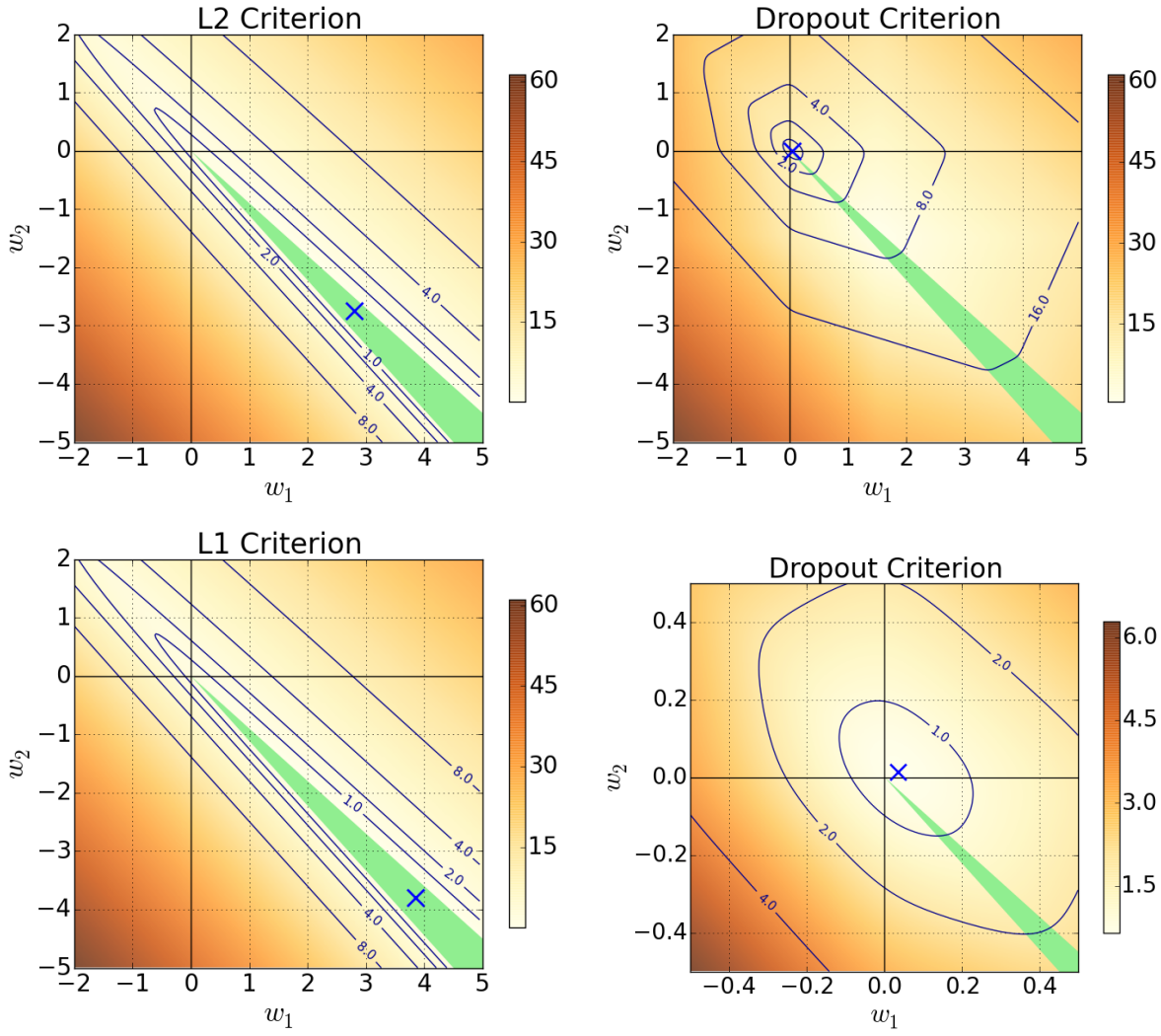


Figure 6: Plots of the criteria and their minimizers for the source P_8. The L_2 and L_1 criteria with $\lambda = 1/100$ are plotted on the right, and the $p = 1/2$ dropout criterion at the same scale and "zoomed in" are shown on the left. As before, the green region indicates the Bayes optimal classifiers.

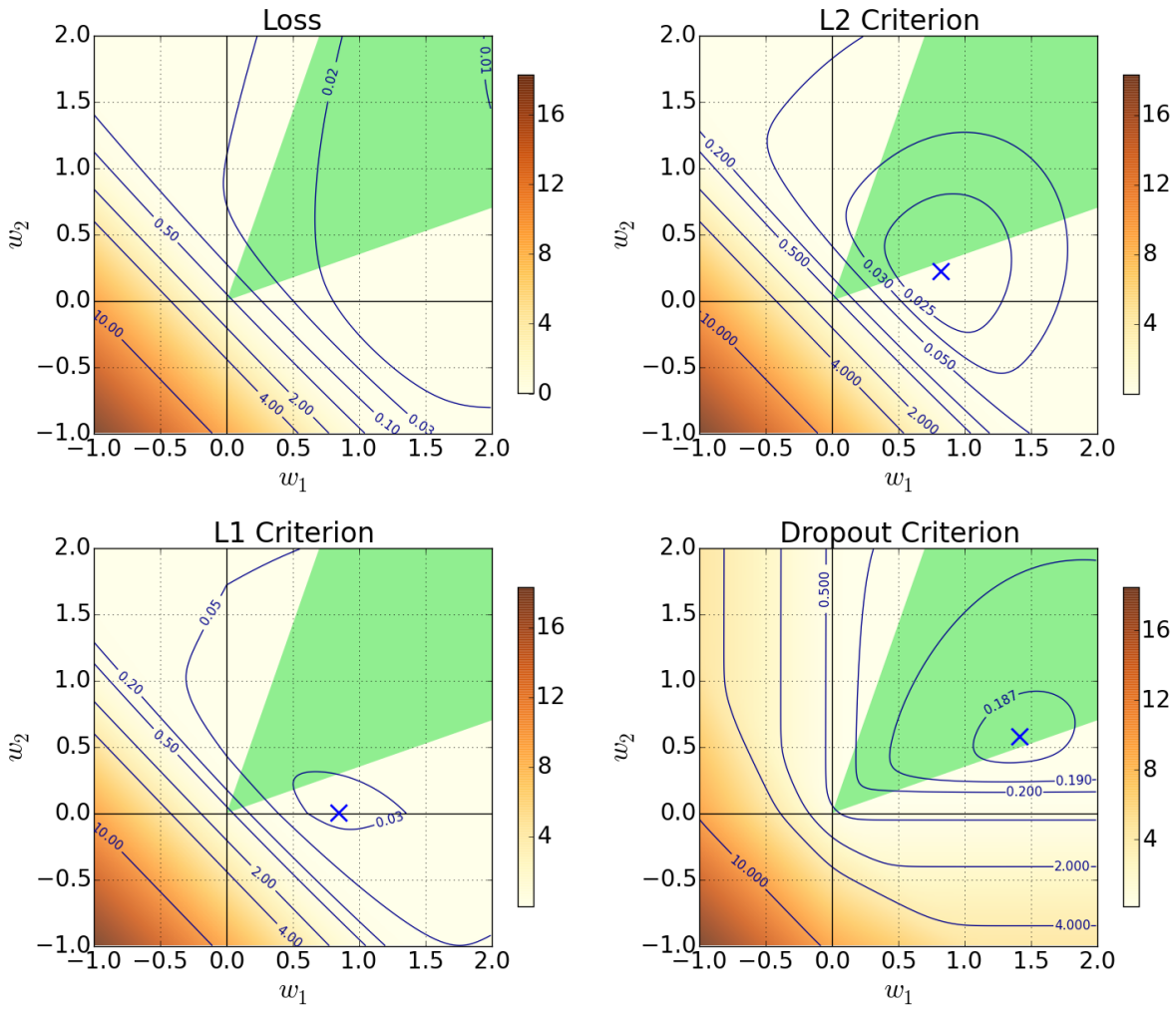


Figure 7: Plots of the loss and various criteria and minimizers for the source P'_8, the modification of P_8 where all the labels are set to 1. As before, $p = 1/2$ for dropout, $\lambda = 1/100$ for the other regularizers, and the green region indicates the Bayes optimal classifiers.

Consider the source P_9 defined as follows. The number n of features is even. All examples are labeled 1. A random example is drawn as follows: the first feature takes the value 1 with probability 9/10 and -1 otherwise, and a subset of exactly $n/2$ of the remaining $n-1$ features (chosen uniformly at random) takes the value 1, and the remaining $n/2-1$ of those first $n-1$ features take the value -1.

A majority vote over the last $n-1$ features achieves perfect prediction accuracy. This is despite the first feature (which does not participate in the vote) being more strongly correlated with the label than any of the voters in the optimal ensemble. Dropout, with its bias for single good features and discrimination against multiple disagreeing features, puts too much weight on this first feature. In contrast, L_2 regularization leads to the Bayes optimal classifier by placing less weight on the first feature than on any of the others.

Theorem 20 *If $\lambda \leq \frac{1}{30n}$ then the weight vector $v(P_9, \lambda)$ optimizing the L_2 criterion has perfect prediction accuracy: $\mathrm{er}_{P_9}(v(P_9, \lambda)) = 0$.*

When $n > 125$, dropout with $q = 1/2$ fails to find the Bayes optimal hypothesis. In particular, we have the following theorem.

Theorem 21 *If the dropout probability $q = 1/2$ and the number of features is an even $n > 125$ then the weight vector $\mathbf{w}^*(P_9, q)$ optimizing the dropout criterion has prediction error rate $\mathrm{er}_{P_9}(\mathbf{w}^*(P_9, q)) \geq 1/10$.*

We conjecture that dropout fails on P_9 for all $n \geq 4$. As evidence, we analyze the $n = 4$ case.

Theorem 22 *If dropout probability $q = 1/2$ and the number of features is $n = 4$ then the minimizer of the dropout criteria $\mathbf{w}^*(P_9, q)$ has has prediction error rate $\mathrm{er}_{P_9}(\mathbf{w}^*(P_9, q)) \geq 1/10$.*

Theorems 20, 21 and 22 are proved in Appendices H, I and J.

10. A High-Dimensional Source Preferred by Dropout

Define the source P_{10}, which depends on (small) positive real parameters η, α, and β, as follows. A random label y is generated first, with both of $+1$ and -1 equally likely. The features $x_1, ..., x_n$ are conditionally independent given y. The first feature tends to be accurate but small: $x_1 = \alpha y$ with probability $1 - \eta$, and is $-\alpha y$ with probability η. The remaining features are larger but less accurate: for $2 \leq i \leq n$, feature x_i is y with probability $1/2 + \beta$, and $-y$ otherwise.

When η is small enough relative to β, the Bayes' optimal prediction is to predict with the first feature. When α is small, this requires concentrating the weight on w_1 to outvote the other features. Dropout is capable of making this one weight large while L_2 regularization is not.

Theorem 23 *If $q = 1/2$, $n \geq 100$, $\alpha > 0$, $\beta = 1/(10\sqrt{n-1})$, and $\eta \leq \frac{1}{2+\exp(54\sqrt{n})}$, then $\mathrm{er}_{P_{10}}(\mathbf{w}^*(P_{10}, q)) = \eta$.*

Theorem 24 *If* $\beta = 1/(10\sqrt{n-1})$, $\lambda = \frac{1}{30n}$, $\alpha < \beta\lambda$, *and* n *is a large enough even number, then for any* $\eta \in [0,1]$, $\mathrm{er}_{P_{10}}(\mathbf{v}(P_{10},\lambda)) \geq 3/10$.

Theorems 23 and 24 are proved in Appendices K and L.

Let $\tilde{n}$ be a large enough even number in the sense of Theorem 24. Let P_η be the distribution defined at the start of Section 10 with number of features $n = \tilde{n}$, $\beta = 1/(10\sqrt{n-1})$, $\alpha = 1/(300n\sqrt{n})$, and $0 < \eta < 1/(2+\exp(54\sqrt{n}))$ is a free parameter. Theorem 23 shows that $\mathrm{er}_{P_\eta}(\mathbf{w}^*(P_\eta, q)) = \eta$ when dropout probability $q = 1/2$. For this same distribution, Theorem 24 shows $\mathrm{er}_{P_\eta}(\mathbf{v}(P_\eta,\lambda)) \geq 3/10$ when $\lambda = 1/30n$. Therefore

$$\frac{\mathrm{er}_{P_\eta}(\mathbf{w}^*(P_\eta, 1/2))}{\mathrm{er}_{P_\eta}(\mathbf{v}(P, 1/30\tilde{n}))}$$

goes to 0 as $\eta \to 0$.

The distribution defined at the start of Section 9, which we call Q here, provides contrasting behavior when $n = \tilde{n}$. Theorem 21 shows that the error $\mathrm{er}_Q(\mathbf{w}^*(Q, 1/2)) \geq 1/10$ while Theorem 20 shows that $\mathrm{er}_Q(v(Q, 1/30\tilde{n}) = 0$. Therefore the P_η and Q distributions strongly separate dropout and L_2 regularization for parameters $q = 1/2$ and $\lambda = 1/30n$.

11. Conclusions

We have built on the interpretation of dropout as a regularizer in Wager et al. (2013) to prove several interesting properties of the dropout regularizer. This interpretation decomposes the dropout criterion minimized by training into a loss term plus a regularization penalty that depends on the feature vectors in the training set (but not the labels). We started with a characterization of when the dropout criterion has a unique minimum, and then turn to properties of the dropout regularization penalty. We verified that the dropout regularization penalty has some desirable properties of a regularizer: it is 0 at the zero vector, and the contribution of each feature vector in the training set is non-negative.

On the other hand, the dropout regularization penalty does not behave like standard regularizers. In particular, we have shown:

1. Although the dropout "loss plus regularization penalty" criterion is convex in the weights $\mathbf{w}$, the regularization penalty imposed by dropout training is *not* convex.

2. Starting from an arbitrary weight vector, any single weight can go to infinity while the dropout regularization penalty remains bounded.

3. In some cases, multiple weights can simultaneously go to infinity while the regularization penalty remains bounded.

4. The regularization penalty can *decrease* as weights increase from 0 when the features are correlated.

These are in stark contrast to standard norm-based regularizers that always diverge as any weight goes to infinity, and are non-decreasing in each individual weight.

In most cases the dropout regularization penalty *does* diverge as multiple weights go to infinity. We characterize when sending two weights to infinity causes the dropout regularization penalty to diverge, and when it will remain finite. In particular, dropout is willing to put a large weights on multiple features if the $w_i x_i$ products tend to have the same sign.

The form of our analytical bounds suggest that the strength of the regularizer grows linearly with the dropout probability q, and provide additional support for the claim (Wager et al., 2013) that dropout favors rare features.

We found it important to check our intuition by working through small examples. To make this more rigorous we needed a definition of when a source favored dropout regularization over a more standard regularizer like L_2. Such a definition needs to deal with the strength of regularization, a difficulty complicated by the fact that dropout regularization is parameterized by the dropout probability $q \in [0, 1]$ while L_2 regularization is parameterized by $\lambda \in [0, \infty]$. Our solution is to consider pairs of sources P and Q. We then say the pair *separates* the dropout and L_2 if dropout with a particular parameter q performs better then L_2 with a particular parameter λ on source P, while L_2 (with the same λ) performs better than dropout (with the same q) on source Q. Our definition uses generalization error as the most natural interpretation of "performs better".

Sections 5 through 10 are devoted to proving that dropout and L_2 are strongly separated by certain pairs of distributions. Section 7 shows that dropout and L_1 regularization are also strongly separated, and Section 8 describes a separation illustrating dropout's bias against co-adaptation of weights. Proving strong separation is non-trivial even after one finds the right distributions. This is due to several factors: the minimizers of the criteria do not have closed forms, we wish to prove separation for ranges of the regularization values, and the binomial distributions induced by dropout are not amenable to exact analysis. Despite these difficulties, the separation results reinforce the intuition that dropout is more willing to use a large weight in order to better fit the training data than L_2 regularization. However, if two features often have both the same and different signs (as in Theorem 13) then dropout is less willing to put even moderate weight on both features.

As a side benefit of these analyses, the plots in Figure 2 and Figure 3 provide a dramatic illustration of the dropout regularizer's non-convexity and its preference for making only a single weight large, and the checkerboard pattern of the dropout regularizer in Figure 5 illustrates its bias against co-adaptation of weights. This is consistent with the insight provided by Theorems 13 and 15.

Some feature transformations appear to have substantially different effects on dropout and L_2. For example, suppose we replace a boolean feature x_i with a batch of features $x_{i,1}, ..., x_{i,k}$, and,

- when $x_i = 0$, we set $x_{i,1} = ... = x_{i,k} = 0$ and
- when $x_i = 1$, we set $x_{i,j'} = 1$ for j' chosen uniformly at random from $\{1, ..., k\}$, and $x_{i,j} = 0$ for $j \neq j'$.

We can think of $x_{i,1}, ..., x_{i,k}$ as a "partition" of x_i. This kind of transformation can arise in document classification when words have alternate spellings, or a single feature representing a set of synonyms is split into features for the individual words (assuming that each document uses only one of the synonyms).

The inductive bias of dropout is apparently not affected by such feature partitioning. For any weight vector $\mathbf{w}$ on the original features, the modified weight vector which copies w_i for each feature in the partition of x_i makes the same predictions and has the same dropout regularization penalty. On the other hand, the L_2 regularization penalty increases. If an algorithm creates k copies of the weight w_i to have the same behavior on the modified data, this increases the penalty arising from this feature by a factor of k, providing an incentive for the algorithm to use other features instead.

Dropout's relative affinity with partitioned features could be another basis of separation with L_2. It suggests that dropout might be able to more effectively exploit rare primitive features, while L_2 regularization benefits from having more frequent higher-level features. This is a potential subject for future research.

Now suppose that, instead of partitioning x_i, we set $x_{i,1}, ..., x_{i,k}$ to be k copies of x_i. In this case, an L_2-regularized algorithm could split weight w_i into k parts, putting weight w_i/k on each copy of x_i. This will classify the transformed data the same way as the original data while *reducing* the L_2 regularization cost of using the feature by a factor of k (since $\sum_j (w_i/k)^2 = (1/k)w_i^2$). Although such feature cloning can also reduce the dropout regularization penalty (see Figure 1 and Proposition 12), we conjecture that the reduction is at most an additive constant.

If this conjecture were true, then L_2-regularized algorithms make heavier use of duplicated features than dropout-regularized algorithms. This in turn suggest that dropout confers resistance to paying undue attention to groups of mostly redundant features. This possibility is another potential subject for future research.

The aim of our analysis has been to aid general understanding of what kinds of problems are well-suited to dropout. A more authoritative idea of whether dropout confers an advantage in a particular case can be gained experimentally.

Linear classifiers are often learned with a bias term, creating a classifier of the form $\text{sign}(\mathbf{w} \cdot \mathbf{x} - b)$. Here the bias b is also learned, but not regularized. We have focused on the case $b = 0$ to keep the analysis simple, and our constructions can be easily modified so that the optimal bias is 0 (see footnote 3). The effect of a non-zero bias term on the general properties in Section 3 can be more subtle, and is a potential subject for future research.

Our analysis is for the logistic regression case corresponding to a single output node. It would be very interesting to have similar analysis for multi-layer neural networks. However, dealing with non-convex loss of such networks will be a major challenge. Another open problem suggested by this work is how the definition of separation can be used to gain insight about other regularizers and settings.

Acknowledgments

We thank the reviewers for their thoughtful comments and corrections.

Appendix A. Proof of Proposition 12

Proposition 12. Fix $p = 1/2$, $w_2 > 0$, and an arbitrary $\mathbf{x} \in (0, \infty)^2$. Let D be the distribution concentrated on $\mathbf{x}$. Then $\mathbf{reg}_{D,q}(w_1, w_2)$ locally decreases as w_1 increases from 0.

First, we show that assuming $\mathbf{x} = (2, 2)$ is without loss of generality. When D concentrates all of its probability on a single $\mathbf{x}$, let us denote $\mathbf{reg}_{D,1/2}$ by $\mathbf{reg}_{\mathbf{x},1/2}$. Since anyplace w_1 appears in the expression for $\mathbf{reg}_{\mathbf{x},1/2}$, it is multiplied by x_1, if we multiply w_1 by some constant c and divide x_1 by c, we do not change $w_1 x_1$, and therefore do not change $\mathbf{reg}_{\mathbf{x},1/2}$. The same holds for w_2. Thus

$$\mathbf{reg}_{\mathbf{x},1/2}(\mathbf{w}) = \mathbf{reg}_{(2,2),1/2}(w_1 x_1/2, w_2 x_2/2).$$

If we change variables and let $\tilde{w}_1 = w_1 x_1/2$ and $\tilde{w}_2 = w_2 x_2/2$, then since x_1 and x_2 are both positive, $\tilde{w}_2$ is positive iff w_2 is, and $\mathbf{reg}_{\mathbf{x},1/2}(\mathbf{w})$ is increasing with w_1 iff $\mathbf{reg}_{(2,2),1/2}(\tilde{\mathbf{w}})$ is increasing with $\tilde{w}_1$.

We continue assuming $\mathbf{x} = (2, 2)$. It suffices to show $\partial \mathbf{reg}_{D,q}(w_1, w_2)/\partial w_1|_{w_1=0} < 0$. This derivative is

$$\frac{3e^{w_2} + e^{-3w_2} - 3e^{-w_2} - e^{3w_2}}{2(e^{w_2} + e^{-w_2})(e^{2w_2} + e^{-2w_2})}. \tag{16}$$

The sign depends only on the numerator, which is 0 when $w_2 = 0$. The derivative of the numerator with respect to w_2 is $3e^{w_2} - 3e^{-3w_2} + 3e^{-w_2} - 3e^{3w_2}$, which is negative for $w_2 > 0$, since $e^z + e^{-z}$ is an increasing function in z. Thus the numerator in (16) is decreasing in w_2. Therefore (16) is negative when $w_2 > 0$, and the regularization penalty is (locally) decreasing as w_1 increases from 0.

(Note: Proposition 12 may be generalized with slight modifications to apply whenever $\mathbf{x}$ has two nonzero components. What is needed is that $x_1 w_1$ and $x_2 w_2$ have the same sign. For example, if x_1 is negative but $x_2 w_2$ is positive, then moving w_1 from 0 in the negative direction decreases $\mathbf{reg}_{D,q}(\mathbf{w})$.)

Appendix B. Proof of Theorem 13

Theorem 13. Fix an arbitrary distribution D with support in $\mathbf{R}^2$, weight vector $\mathbf{w} \in \mathbf{R}^2$, and non-dropout probability p. If there is an $\mathbf{x}$ with positive probability under D such that $w_1 x_1$ and $w_2 x_2$ are both non-zero and have different signs, then the regularization penalty $\mathbf{reg}_{D,q}(\omega \mathbf{w})$ goes to infinity as ω goes to $\pm\infty$.

Fix an $\mathbf{x}$ satisfying the conditions of the theorem.

$$\begin{aligned}
\mathbf{reg}_{D,q}(\omega \mathbf{w}) &\geq D(\mathbf{x}) \mathbf{E}_{\boldsymbol{\nu}} \left(\ln \left(\frac{\exp(-\frac{\omega \mathbf{w} \cdot (\mathbf{x} + \boldsymbol{\nu})}{2}) + \exp(\frac{\omega \mathbf{w} \cdot (\mathbf{x} + \boldsymbol{\nu})}{2})}{\exp(\frac{-\omega \mathbf{w} \cdot \mathbf{x}}{2}) + \exp(\frac{\omega \mathbf{w} \cdot \mathbf{x}}{2})} \right) \right) \\
&> D(\mathbf{x}) \mathbf{E}_{\boldsymbol{\nu}} \left(\ln \left(\frac{\exp(\frac{|\omega \mathbf{w} \cdot (\mathbf{x} + \boldsymbol{\nu})|}{2})}{2 \exp(\frac{|\omega \mathbf{w} \cdot \mathbf{x}|}{2})} \right) \right) \\
&= D(\mathbf{x}) \mathbf{E}_{\boldsymbol{\nu}} \left(-\ln(2) + \frac{|\omega \mathbf{w} \cdot (\mathbf{x} + \boldsymbol{\nu})|}{2} - \frac{|\omega \mathbf{w} \cdot \mathbf{x}|}{2} \right).
\end{aligned} \tag{17}$$

We now examine the expectation over $\boldsymbol{\nu}$ of the term that depends on $\boldsymbol{\nu}$. We assume that $|w_1x_1| \geq |w_2x_2|$ so $|\mathbf{w}\cdot\mathbf{x}| = |w_1x_1| - |w_2x_2|$; the other case is symmetrical.

$$\begin{aligned}\mathbf{E}_{\boldsymbol{\nu}}(|\omega\mathbf{w}\cdot(\mathbf{x}+\boldsymbol{\nu})|) &= |\omega|\big(p^2|\mathbf{w}\cdot\mathbf{x}/p| + p(1-p)|w_1x_1/p| + p(1-p)|w_2x_2/p|\big)\\ &= |\omega|\big(p|\mathbf{w}\cdot\mathbf{x}| + (1-p)(|w_1x_1| - |w_2x_2| + |w_2x_2|) + (1-p)|w_2x_2|\big)\\ &= |\omega|(|\mathbf{w}\cdot\mathbf{x}| + 2(1-p)|w_2x_2|).\end{aligned}$$

Plugging this into (17) gives:

$$\mathbf{reg}_{D,q}(\omega\mathbf{w}) > D(\mathbf{x})\,(-\ln 2 + (1-p)|\omega||w_2x_2|)$$

which goes to infinity as ω goes to $\pm\infty$.

Appendix C. Proof of Theorem 15

Theorem 15. Let $\mathbf{w}$ be a weight vector and D be a discrete distribution such that $w_ix_i \geq 0$ for each index i and all $\mathbf{x}$ in the support of D. The limit of $\mathbf{reg}_{D,q}(\omega\mathbf{w})$ as ω goes to infinity is bounded by $\ln(2)(1-p)\mathbf{P}_{\mathbf{x}\sim D}(\mathbf{w}\cdot\mathbf{x}\neq 0)$.

First note that If $\mathbf{w}$ and D are such that $\mathbf{w}\cdot\mathbf{x} = 0$ for all $\mathbf{x}$ in the support of D, then $\mathbf{reg}_{D,q}(\mathbf{w}) = \mathbf{reg}_{D,q}(\omega\mathbf{w}) = 0$. We now analyze the general case.

$$\begin{aligned}\mathbf{reg}_{D,q}(\omega\mathbf{w}) &= \mathbf{E}_{\mathbf{x},\boldsymbol{\nu}}\left(\ln\left(\frac{\exp(\frac{\omega\mathbf{w}\cdot(\mathbf{x}+\boldsymbol{\nu})}{2}) + \exp(\frac{-\omega\mathbf{w}\cdot(\mathbf{x}+\boldsymbol{\nu})}{2})}{\exp(\frac{\omega\mathbf{w}\cdot\mathbf{x}}{2}) + \exp(\frac{-\omega\mathbf{w}\cdot\mathbf{x}}{2})}\right)\right)\\ &= \mathbf{E}_{\mathbf{x},\boldsymbol{\nu}}\left(\ln\left(\frac{\exp(\frac{\omega\mathbf{w}\cdot(\mathbf{x}+\nu)}{2})(1+\exp(-\omega\mathbf{w}\cdot(\mathbf{x}+\boldsymbol{\nu})))}{\exp(\frac{\omega\mathbf{w}\cdot\mathbf{x}}{2})(1+\exp(-\omega\mathbf{w}\cdot\mathbf{x}))}\right)\right)\\ &= \mathbf{E}_{\mathbf{x},\boldsymbol{\nu}}\Big(\,(\omega\mathbf{w}\cdot(\mathbf{x}+\boldsymbol{\nu})/2) + \ln\left(1+\exp\left(-\omega\mathbf{w}\cdot(\mathbf{x}+\boldsymbol{\nu})\right)\right)\\ &\qquad -(\omega\mathbf{w}\cdot\mathbf{x}/2) - \ln\left(1+\exp\left(-\omega\mathbf{w}\mathbf{x}\right)\right)\Big).\end{aligned} \tag{18}$$

Of the four terms inside the expectation in Equation (18), the first and third cancel since the expectation of $\boldsymbol{\nu}$ is $\mathbf{0}$. Therefore:

$$\mathbf{reg}_{D,q}(\omega\mathbf{w}) = \mathbf{E}_{\mathbf{x}}\big(\mathbf{E}_{\boldsymbol{\nu}}\big(\ln(1+\exp(-\omega\mathbf{w}\cdot(\mathbf{x}+\boldsymbol{\nu}))) - \ln(1+\exp(-\omega\mathbf{w}\mathbf{x}))\big)\big). \tag{19}$$

Define $\mathrm{nez}(\mathbf{w},\mathbf{x})$ to be the number of indices i where $w_ix_i \neq 0$. We now consider cases based on $\mathrm{nez}(\mathbf{w},\mathbf{x})$.

Whenever $\mathrm{nez}(\mathbf{w},\mathbf{x}) = 0$ then both $\mathbf{w}\cdot\mathbf{x} = 0$ and $\mathbf{w}\cdot(\mathbf{x}+\boldsymbol{\nu}) = 0$. Therefore the contribution of these $\mathbf{x}$ to the expectation in (19) is $\ln(2) - \ln(2) = 0$.

If $\mathrm{nez}(\mathbf{w},\mathbf{x}) > 0$ then $\mathbf{w}\cdot\mathbf{x} > 0$ (since each $w_ix_i \geq 0$), and the second term of (19) goes to zero as ω goes to infinity. The first term of (19) also goes to zero, *unless* all of the $\mathrm{nez}(\mathbf{w},\mathbf{x})$ components where $w_ix_i > 0$ are dropped out. If they are all dropped out, then the first term becomes $\ln(2)$. The probability that all $\mathrm{nez}(\mathbf{w},\mathbf{x})$ non-zero components are

simultaneously dropped out is $(1-p)^{\text{nez}(\mathbf{w},\mathbf{x})}$. With this reasoning we get from (19) that:

$$\begin{aligned}
&\lim_{\omega\to\infty} \mathbf{reg}_{D,q}(\omega\mathbf{w}) \\
&= \sum_{k=1}^{n} \mathbf{P}_{\mathbf{x}\sim D}(\text{nez}(\mathbf{w},\mathbf{x}) = k)\left(\ln(2)(1-p)^k\right) \qquad (20)\\
&\leq \sum_{k=1}^{n} \mathbf{P}_{\mathbf{x}\sim D}(\text{nez}(\mathbf{w},\mathbf{x}) = k)\left(\ln(2)(1-p)\right) \\
&= \ln(2)(1-p)\mathbf{P}(\mathbf{w}\cdot\mathbf{x}\neq 0)
\end{aligned}$$

as desired.

(Note that Equation 20 gives a precise, but more complex expression for the limit.)

Appendix D. Proof of Theorem 16

Theorem 16. If $0 < \lambda \leq 1/50$, then $\text{er}_{P_5}(\mathbf{v}(P_5,\lambda)) = 0$ for the distribution P_5 defined in (12).

To keep the notation clean let us abbreviate P_5 as just P throughout this proof.

By scaling the L_2 criterion we can obtain cancellation in the expectation. Let $\mathbf{v}$ be weight vector found by minimizing the following L_2 regularized criterion J:

$$J(\mathbf{w}) = 3\left(\mathbf{E}_{(\mathbf{x},y)\sim P}(\ell(y(\mathbf{w}\cdot\mathbf{x}))) + (\lambda/2)||\mathbf{w}||^2\right). \qquad (21)$$

Note the factor of 3 is to simplify the expressions and doesn't affect the minimizing $\mathbf{v}$.

We will prove Theorem 16 with a series of lemmas.

But first, let's take some partial derivatives:

$$\frac{\partial J}{\partial w_1} = \frac{-10}{1+\exp(10w_1 - w_2)} + \frac{-1.1}{1+\exp(1.1w_1 - w_2)} + \frac{1}{1+\exp(-w_1 + 1.1w_2)} + 3\lambda w_1 \qquad (22)$$

$$\frac{\partial J}{\partial w_2} = \frac{1}{1+\exp(10w_1 - w_2)} + \frac{1}{1+\exp(1.1w_1 - w_2)} + \frac{-1.1}{1+\exp(-w_1 + 1.1w_2)} + 3\lambda w_2. \qquad (23)$$

We will repeatedly use the following basic, well-known, lemma.

Lemma 25 *For any convex, differentiable function ψ defined on $\mathbf{R}^n$ with a unique minimum $\mathbf{w}^*$, for any $\mathbf{w}\in\mathbf{R}^n$, if $g(\mathbf{w})$ is the gradient of ψ at $\mathbf{w}$ then $\mathbf{w}^*$ is contained in the closed halfspace whose separating hyperplane goes through $\mathbf{w}$, and whose normal vector is $-g(\mathbf{w})$; i.e., $\mathbf{w}^*\cdot g(\mathbf{w}) \leq \mathbf{w}\cdot g(\mathbf{w})$. Furthermore, if $g(\mathbf{w})\neq\mathbf{0}$ then $\mathbf{w}^*\cdot g(\mathbf{w}) < \mathbf{w}\cdot g(\mathbf{w})$.*

Now we're ready to start our analysis of P.

Lemma 26 *If $0\leq\lambda$, the optimizing v_1 is positive.*

Proof: By Lemma 25, it suffices to show that there is a point $(0, a_2)$ where both $\frac{\partial J}{\partial w_1}\big|_{(0,a_2)} < 0$ and $\frac{\partial J}{\partial w_2}\big|_{(0,a_2)} = 0$.

From Equation (22):

$$\left.\frac{\partial J}{\partial w_1}\right|_{(0,a_2)} = \frac{-11.1}{1+\exp(-a_2)} + \frac{1}{1+\exp(1.1a_2)}$$

and each term is decreasing as a_2 increases. Since it is negative when $a_2 = -2$, we have $\left.\frac{\partial J}{\partial w_1}\right|_{(0,a_2)} < 0$ for all $a_2 > -2$. So, to prove the lemma, if suffices to show that there is a $a_2 \in (-2, \infty)$ such that the other derivative $\left.\frac{\partial J}{\partial w_2}\right|_{(0,a_2)} = 0$.

From equation (23):

$$\left.\frac{\partial J}{\partial w_2}\right|_{(0,a_2)} = \frac{2}{1+\exp(-a_2)} + \frac{-1.1}{1+\exp(1.1a_2)} + 3\lambda a_2$$

and each term is continuously increasing in a_2. When $a_2 = -2$, $\left.\frac{\partial J}{\partial w_2}\right|_{(0,a_2)}$ is negative. On the other hand, $\frac{\partial J}{\partial w_2}\big|_{(0,0)}$ is positive. Therefore for some $a_2 \in (-2, 0)$ we have $\frac{\partial J}{\partial w_2}\big|_{(0,a_2)} = 0$ as desired. ∎

Lemma 27 *There is a real $a > 0$ such that*

$$\left.\frac{\partial J(\mathbf{w})}{\partial w_1}\right|_{(a,a)} + \left.\frac{\partial J(\mathbf{w})}{\partial w_2}\right|_{(a,a)} = 0.$$

Proof: Applying (22) and (23), we get

$$b \stackrel{\text{def}}{=} \left.\frac{\partial J(\mathbf{w})}{\partial w_1}\right|_{(a,a)} + \left.\frac{\partial J(\mathbf{w})}{\partial w_2}\right|_{(a,a)} = \frac{-9}{1+\exp(9a)} + \frac{-0.2}{1+\exp(a/10)} + 6\lambda a.$$

Since b is negative when $a = 0$ and is a continuous function of a, and $\lim_{a\to\infty} b > \infty$, the lemma holds. ∎

Lemma 28 $v_1 \geq v_2$.

Proof: Let a be the value from Lemma 27, and let $\mathbf{g} = (g_1, g_2)$ be the gradient of J at (a, a). Lemma 25 implies that $\mathbf{v}$ lies in the halfspace through (a, a) in the direction of $-\mathbf{g}$. Lemma 27 implies that

$$g_1 = \left.\frac{\partial J(\mathbf{w})}{\partial w_1}\right|_{(a,a)} = -\left.\frac{\partial J(\mathbf{w})}{\partial w_2}\right|_{(a,a)} = -g_2.$$

Examination of the derivatives (22) and (23) at (a, a) shows that the first term of (22) is negative and the first term of (23) is positive while the last three terms match (although

in a different order). Therefore $g_1 < 0$ and $g_2 = -g_1$ is positive. Applying Lemma 25 completes the proof. ∎

Lemma 28 implies that $\mathbf{v}$ correctly classifies $(10, -1)$ and $(11/10, -1)$. It remains to show that $\mathbf{v}$ correctly classifies $(-1, 11/10)$, that is, that v_1 is not *too* much bigger than v_2.

Lemma 29 *If $v_2 \geq 0.6$ and $\lambda > 0$ then $v_1 < 11v_2/10$.*

Proof: Combining $\frac{\partial J}{\partial w_1}\Big|_{\mathbf{v}} = 0$ with (22), we get

$$3\lambda v_1 = \frac{10}{1+\exp(10v_1 - v_2)} + \frac{1.1}{1+\exp(1.1v_1 - v_2)} + \frac{-1}{1+\exp(-v_1 + 1.1v_2)}$$

and, similarly,

$$3\lambda v_2 = \frac{-1}{1+\exp(10v_1 - v_2)} + \frac{-1}{1+\exp(1.1v_1 - v_2)} + \frac{1.1}{1+\exp(-v_1 + 1.1v_2)}.$$

Thus

$$3\lambda(10v_1 - 11v_2) = \frac{111}{1+\exp(10v_1 - v_2)} + \frac{22}{1+\exp(1.1v_1 - v_2)} - \frac{22.1}{1+\exp(-v_1 + 1.1v_2)}. \quad (24)$$

Assume for contraction that $v_1 \geq 11v_2/10$. Then $10v_1 - v_2 \geq 10v_2$, $1.1v_1 - v_2 \geq 0.21v_2$, and $-v_1 + 1.1v_2 \leq 0$, so

$$3\lambda(10v_1 - 11v_2) \leq \frac{111}{1+\exp(10v_2)} + \frac{22}{1+\exp(0.21v_2)} - 11.05.$$

However, $10v_1 - 11v_2 \geq 0$ and (since $v_2 \geq 0.6$) the RHS is negative, giving the desired contradiction. ∎

Lemma 30 *If $0 < \lambda \leq 1/50$ then $v_2 \geq 0.6$.*

Proof: It suffices to show that there is a point $(x, 0.6)$ where the partial w.r.t. w_1 is 0 and the partial w.r.t w_2 is negative.

$$\frac{\partial J}{\partial w_1}\Big|_{(x,0.6)} = \frac{-10}{1+\exp(10x - 0.6)} + \frac{-1.1}{1+\exp(1.1x - 0.6)} + \frac{1}{1+\exp(-x + 0.66)} + 3\lambda x$$

and is increasing in x and λ (assuming $x > 0$) and becomes positive as x goes to infinity. It is negative when evaluated at $x = 0.6$ and $\lambda = 1/50$, so for all $\lambda \leq 1/50$ there is an $x > 0.6$ such that $\partial J/\partial w_+\big|_{(x,1)} = 0$.

$$\frac{\partial J}{\partial w_2}\Big|_{(x,0.6)} = \frac{1}{1+\exp(10x - 0.6)} + \frac{1}{1+\exp(1.1x - 0.6)} + \frac{-1.1}{1+\exp(-x + 0.66)} + 1.8\lambda$$

and is decreasing in x and increasing in λ. It is negative when $x = 0.6$ and $\lambda = 1/50$, so it will remain negative for all $x > 0.6$ and $0 \leq \lambda \leq 1/50$, as desired. ∎

So, we have shown that, if $\lambda \leq 1/50$, then all examples are classified correctly by $\mathbf{v}$, which proves Theorem 16.

Appendix E. Proof of Theorem 17

Theorem 17. If $q \geq 1/3$ then $\mathrm{er}_{P_5}(\mathbf{w}^*(P_5, q)) \geq 1/3$ for the distribution P_5 defined in (12).

Throughout this proof we also abbreviate P_5 as just P.

For this subsection, let us define the scaled dropout criterion

$$J(\mathbf{w}) = 3\, \mathbf{E}_{(\mathbf{x},y)\sim P,\mathbf{r}}(\ell(y(\mathbf{w} \cdot (\mathbf{r} \odot \mathbf{x})))) \tag{25}$$

where the components of $\mathbf{r}$ are independent samples from a Bernoulli distribution with parameter $p = 1 - q > 0$. Again, the factor of 3 is to simplify the expectation and doesn't change the minimizing $\mathbf{w}$. Let $\mathbf{w}^{\circledast}$ be the minimizer of this $J(\mathbf{w})$, so that Equation (4) implies that the optimizer $\mathbf{w}^*$ of the dropout criterion is $p\mathbf{w}^{\circledast}$. Note that $\mathbf{w}^*$ classifies an example correctly if and only if $\mathbf{w}^{\circledast}$ does.

Next, note that we may assume without loss of generality that both components of $\mathbf{w}^{\circledast}$ are positive, since, if either is negative, one of $(-1, 1.1)$ or $(1.1, -1)$ is misclassified and we are done.

We will prove Theorem 17 by proving that, when $q \geq 1/3$, $\mathbf{w}^{\circledast}$ misclassifies $(-1, 1.1)$, or, equivalently, that $w_1^{\circledast} > (11/10)w_2^{\circledast}$.

First, let us evaluate some partial derivatives. (Note that, if x_i is dropped out, the value of w_i does not matter.)

$$\frac{\partial J}{\partial w_1} = (1-q)^2 \left(\frac{-10}{1+\exp(10w_1 - w_2)} + \frac{-1.1}{1+\exp(1.1w_1 - w_2)} + \frac{1}{1+\exp(-w_1 + 1.1w_2)} \right) \tag{26}$$

$$+ (1-q)q \left(\frac{-10}{1+\exp(10w_1)} + \frac{-1.1}{1+\exp(1.1w_1)} + \frac{1}{1+\exp(-w_1)} \right)$$

$$\frac{\partial J}{\partial w_2} = (1-q)^2 \left(\frac{1}{1+\exp(10w_1 - w_2)} + \frac{1}{1+\exp(1.1w_1 - w_2)} + \frac{-1.1}{1+\exp(-w_1 + 1.1w_2)} \right) \tag{27}$$

$$+ q(1-q) \left(\frac{1}{1+\exp(-w_2)} + \frac{1}{1+\exp(-w_2)} + \frac{-1.1}{1+\exp(1.1w_2)} \right).$$

The following is the key lemma. As before, it is useful since, for any $\mathbf{w}$, if $g(\mathbf{w})$ is nonzero, then $\mathbf{w}^{\circledast}$ lies in the open halfspace through $\mathbf{w}$ whose normal vector is the negative gradient.

Lemma 31 *For all $a > 0$ and $q \geq 1/3$,*

$$\left.\frac{\partial J}{\partial w_2}\right|_{(a,10a/11)} > 0. \tag{28}$$

Proof: We have

$$\left.\frac{\partial J}{\partial w_2}\right|_{(a,10a/11)} = (1-q)^2 \left(\frac{1}{1+\exp(100a/11)} + \frac{1}{1+\exp(21a/110)} + \frac{-1.1}{2} \right)$$

$$+ q(1-q) \left(\frac{2}{1+\exp(-10a/11)} + \frac{-1.1}{1+\exp(a)} \right).$$

Note that this derivative is positive if and only if

$$\begin{aligned}
&f(q,a)\\
&= \left(\frac{1}{1-q}\right) \left.\frac{\partial J}{\partial w_2}\right|_{(a,10a/11)}\\
=&q\left(\frac{11}{20} + \frac{2}{1+\exp(-10a/11)} + \frac{-1}{1+\exp(21a/110)} + \frac{-1}{1+\exp(100a/11)} + \frac{-11/10}{1+\exp(a)}\right)\\
&+ \frac{1}{1+\exp(21a/110)} + \frac{1}{1+\exp(100a/11)} + \frac{-11}{20}
\end{aligned}$$

is positive, as $0 < q < 1$. Note that the terms multiplying q are increasing in a and sum to 0 when $a = 0$. On the other hand, the terms not multiplied by q are decreasing in a and turn negative when a is just over $1/4$. Thus both parts are positive when $a \leq 1/4$. Note that $f(q,a)$ can be underestimated by underestimating a on the q-terms and overestimating a on the other terms.

For $1/4 \leq a \leq 2$,

$$\begin{aligned}
&f(q,a)\\
&\geq q\left(\frac{11}{20} + \frac{2}{1+\exp(-10/44)} + \frac{-1}{1+\exp(21/440)} + \frac{-1}{1+\exp(100/44)} + \frac{-11/10}{1+\exp(1/4)}\right)\\
&+ \frac{1}{1+\exp(42/110)} + \frac{1}{1+\exp(200/11)} + \frac{-11}{20}\\
&\geq 0.5q - 0.15
\end{aligned}$$

and is positive whenever $q \geq 1/3$.

For $a \geq 2$,

$$\begin{aligned}
&f(q,a)\\
&\geq q\left(\frac{11}{20} + \frac{2}{1+\exp(-20/11)} + \frac{-1}{1+\exp(42/110)} + \frac{-1}{1+\exp(200/11)} + \frac{-11/10}{1+\exp(2)}\right) + \frac{-11}{20}\\
&\geq 1.7q - 11/20
\end{aligned}$$

and is also positive whenever $q \geq 1/3$. ∎

Proof of Theorem 17: Let $\mathbf{g} = (g_1, g_2)$ be the gradient of J at $(w_1^{\circledast}, 10w_1^{\circledast}/11)$. Lemma 31 shows $\mathbf{g}$ is not $\mathbf{0}$, so by convexity

$$\mathbf{w}^{\circledast} \cdot \mathbf{g} < (w_1^{\circledast}, 10w_1^{\circledast}/11) \cdot \mathbf{g}$$

which implies

$$w_2^{\circledast}\, g_2 < (10w_1^{\circledast}/11)\, g_2.$$

Since $g_2 > 0$ (Lemma 31), this implies

$$w_2^{\circledast} < (10w_1^{\circledast}/11)$$

and the $(-1, 11/10)$ example is misclassified by $\mathbf{w}^{\circledast}$, and therefore by $\mathbf{w}^*$, completing the proof. ∎

Appendix F. Proof of Theorem 18

Theorem 18. If $1/100 \leq \lambda \leq 1$, then $\mathrm{er}_{P_6}(\mathbf{v}(P_6, \lambda)) \geq 1/7$ for the distribution P_6 defined in (13).

To keep the notation clean, in this section let us abbreviate P_6 simply as P.

As the reader might expect, we will prove Theorem 18 by proving that $\mathbf{v}$ fails to correctly classify $(1/10, -1)$, that is, by proving that $v_1 < 10v_2$.

We may assume that $v_1 > 0$, since, otherwise, $(1, 0)$ is misclassified.

To obtain cancellation in the expectation, we work with the scaled L_2 criterion

$$J(\mathbf{w}) = 7\mathbf{E}_{(\mathbf{x},y)\sim P}(\ell(y(\mathbf{w}\cdot\mathbf{x}))) + (7\lambda/2)||\mathbf{w}||^2. \tag{29}$$

and let $\mathbf{v}(P, \lambda)$ be the vector minimizing this J, which we often abbreviate as simply $\mathbf{v}$, leaving it implicitly a function of λ. Note that this scaling of the criteria does not change the minimizing $\mathbf{v}$.

Taking derivatives,

$$\frac{\partial J}{\partial w_1} = \frac{-3}{1+\exp(w_1)} + \frac{3\epsilon}{1+\exp(-\epsilon w_1 + w_2)} + \frac{-0.1}{1+\exp(w_1/10 - w_2)} + 7\lambda w_1 \tag{30}$$

$$\frac{\partial J}{\partial w_2} = \frac{-3}{1+\exp(-\epsilon w_1 + w_2)} + \frac{1}{1+\exp(w_1/10 - w_2)} + 7\lambda w_2. \tag{31}$$

Lemma 32 *If either:* $\lambda \geq 1/100$ *and* $a \geq 1/3$*, or* $\lambda \geq 1/4$ *and* $a \geq 1/15$ *then*

$$\left.\frac{\partial J(\mathbf{w})}{\partial w_1}\right|_{(10a,a)} > 0.$$

Proof: We have

$$\begin{aligned}
&\left.\frac{\partial J(\mathbf{w})}{\partial w_1}\right|_{(10a,a)} \\
&= \frac{-3}{(1+\exp(10a))} + \frac{3\epsilon}{1+\exp((1-10\epsilon)a)} + \frac{-1}{20} + 70\lambda a \\
&> \frac{-3}{(1+\exp(10a))} + \frac{-1}{20} + 70\lambda a.
\end{aligned}$$

Each term of the RHS is non-decreasing in a and λ, and the RHS is positive when either $\lambda = 1/100$ and $a = 1/3$ or $\lambda = 1/4$ and $a = 1/15$. ∎

To apply this, we want to show that v_2 is large enough, which we do next.

Lemma 33 *If* $\lambda \leq 1/4$ *then* $v_2 \geq 1/3$ *and if* $\lambda \leq 1$ *then* $v_2 \geq 1/15$.

Proof: Assume to the contrary that $\lambda \leq 1/4$ but $v_2 < 1/3$. From (31), and using that $v_1 > 0$, we have

$$\left.\frac{\partial J}{\partial w_2}\right|_{\mathbf{v}} < \frac{-3}{1+\exp(v_2)} + \frac{1}{1+\exp(-v_2)} + 7\lambda v_2, \tag{32}$$

$x_1 r_1$	$x_2 r_2$	y	seven times probability			$\mathbf{w}^{\circledast} \cdot (\mathbf{r} \odot \mathbf{x})$ over-estimate
0	0	1	$3q$	$+3q^2$	$+q^2$	0
1	0	1	$3(1-q)$			∞
0	1	1		$3q(1-q)$		w_2
$-1/1000$	0	1		$3q(1-q)$		0
$-1/1000$	1	1		$3(1-q)^2$		w_2
0	-1	1			$q(1-q)$	∞
$1/10$	0	1			$q(1-q)$	∞
$1/10$	-1	1			$(1-q)^2$	∞

Table 2: Seven times the dropout distribution. The three probability sub-columns correspond to the original examples (1,0), (-1/1000, 1), (1/10, -1), and the final column is the over-estimate used in Lemma 36.

a bound that is increasing in v_2 and λ. Since $\frac{\partial J}{\partial w_2}\Big|_{\mathbf{v}} = 0$, the bound must be positive. However, when $v_2 \leq 1/3$ and $\lambda \leq 1/4$, it is negative, giving the desired contradiction.

Since the bound (32) is also negative at $v_2 = 1/15$ and $\lambda = 1$, a similar contradiction proves the other half of the lemma. ∎

Proof: (of Theorem 18): Lemmas 32 and 33 imply that $(10v_2, v_2)$ is not the minimizing $\mathbf{v}$ (when $\lambda \geq 1/100$), so by convexity,

$$J(10v_2, v_1) + \big((v_1, v_2) - (10v_2, v_2)\big) \cdot \nabla J(10v_2, v_2) < J(v_1, v_2) \tag{33}$$

$$(v1 - 10v_2) \left.\frac{\partial J}{\partial w_2}\right|_{(10v_2, v_2)} < 0. \tag{34}$$

If $1/100 \leq \lambda \leq 1/4$ then Lemma 33 shows that $v_2 \geq 1/3$ and if $1/4 \leq \lambda \leq 1$ then it shows that $v_2 \geq 1/15$. In either case, Lemma 32 shows that that $\left.\frac{\partial J}{\partial w_2}\right|_{(10v_2, v_2)} > 0$. Therefore,

$$v_1 < 10v_2$$

and $(0.1, -1)$ is misclassified by $\mathbf{v}$, completing the proof. ∎

Appendix G. Proof of Theorem 19

Theorem 19. If $q \leq 1/2$, then $\mathrm{er}_{P_6}(\mathbf{w}^*(P_6, q)) = 0$ for the distribution P_6 defined in (13).

In this proof, let us abbreviate P_6 with just P, and use ϵ to denote 1/1000.

For this section, let us define the scaled dropout criterion

$$J(\mathbf{w}) = 7\mathbf{E}_{(\mathbf{x},y)\sim P, \mathbf{r}}(\ell(y(\mathbf{w} \cdot (\mathbf{r} \odot \mathbf{x})))), \tag{35}$$

where, as earlier, the components of $\mathbf{r}$ are independent samples from a Bernoulli distribution with parameter $p = 1 - q = 1/2 > 0$. (Note that, similarly to before, scaling up the objective

function by 7 does not change the minimizer of J.) See Table 2 for a tabular representation of the distribution after dropout. Let $\mathbf{w}^{\circledast}$ be the minimizer of J, so that $\mathbf{w}^* = p\mathbf{w}^{\circledast}$ (see Equation (4)).

First, let us evaluate some partial derivatives (note that $1-q = (1-q)^2 + q(1-q)$).

$$\begin{aligned}\frac{\partial J}{\partial w_1} &= (1-q)^2\left(\frac{-3}{1+\exp(w_1)} + \frac{3\epsilon}{1+\exp(-\epsilon w_1 + w_2)} + \frac{-0.1}{1+\exp(0.1w_1 - w_2)}\right) \\ &\quad + (1-q)q\left(\frac{-3}{1+\exp(w_1)} + \frac{3\epsilon}{1+\exp(-\epsilon w_1)} + \frac{-0.1}{1+\exp(0.1w_1)}\right)\end{aligned} \tag{36}$$

$$\begin{aligned}\frac{\partial J}{\partial w_2} &= (1-q)^2\left(\frac{-3}{1+\exp(-\epsilon w_1 + w_2)} + \frac{1}{1+\exp(0.1w_1 - w_2)}\right) \\ &\quad + q(1-q)\left(\frac{-3}{1+\exp(w_2)} + \frac{1}{1+\exp(-w_2)}\right).\end{aligned} \tag{37}$$

Let's get started by showing that $\mathbf{w}^{\circledast}$ correctly classifies $(1,0)$.

Lemma 34 $w_1^{\circledast} > 0$.

Proof: As before, it suffices to show that there is a point $(0,a_2)$ where both $\frac{\partial J}{\partial w_1}\big|_{(0,a_2)} < 0$ and $\frac{\partial J}{\partial w_2}\big|_{(0,a_2)} = 0$.

From Equation (36):

$$\frac{\partial J}{\partial w_1}\Big|_{(0,a_2)} = (1-q)^2\left(\frac{-3}{2} + \frac{3\epsilon}{1+\exp(a_2)} + \frac{-0.1}{1+\exp(-a_2)}\right) + \frac{(1-q)q}{2}\left(-3.1 + 3\epsilon\right)$$

which is decreasing in a_2, and negative even as a_2 approaches $-\infty$ (recalling $\epsilon = 1/1000$), so $\frac{\partial J}{\partial w_1}\Big|_{(0,a_2)}$ is always negative.

Equation (37) implies

$$\begin{aligned}&\frac{\partial J}{\partial w_2}\Big|_{(0,a_2)} \\ &= (1-q)^2\left(\frac{-3}{1+\exp(a_2)} + \frac{1}{1+\exp(-a_2)}\right) + q(1-q)\left(\frac{-3}{1+\exp(a_2)} + \frac{1}{1+\exp(-a_2)}\right).\end{aligned}$$

This is negative when $a_2 = 0$, approaches $1-q$ as a_2 goes to infinity, and is continuous, so there is a a_2 such that $\frac{\partial J}{\partial w_2}\Big|_{(0,a_2)} = 0$. Since $\frac{\partial J}{\partial w_1}\Big|_{(0,a_2)} < 0$, this proves the lemma. ∎

Next, we'll start to work on showing that $\mathbf{w}^{\circledast}$ correctly classifies $(-\epsilon, 1)$.

Lemma 35 *For all $a > 1/10$,*

$$\frac{\partial J}{\partial w_1}\Big|_{(a/\epsilon,a)} > 0.$$

Proof: From (36), we have

$$\begin{aligned}\frac{\partial J}{\partial w_1}\Big|_{(a/\epsilon,a)} =& (1-q)^2\left(\frac{-3}{1+\exp(a/\epsilon)} + \frac{3\epsilon}{1+\exp(0)} + \frac{-0.1}{1+\exp(0.1(a/\epsilon) - a)}\right) \\ &+ q(1-q)\left(\frac{-3}{1+\exp(a/\epsilon)} + \frac{3\epsilon}{1+\exp(-a)} + \frac{-0.1}{1+\exp(a/10\epsilon)}\right)\end{aligned}$$

which is positive if $a > 1/10$ as the positive terms (even with the ϵ factors) dominate the negative ones. ∎

Lemma 36

$$w_2^{\circledast} > 1/4.$$

Proof: Assuming $w_1 \geq 0$, the estimates in Table 2 along with the facts that $\ell(z)$ is positive and decreasing show :

$$J(\mathbf{w}) \geq 3(1-q)\ln(1+\exp(-w_2)) + 6q\ln(2) + q^2\ln(2) \tag{38}$$

which is decreasing in w_2. If $w_2^{\circledast} \leq 1/4$, then bound (38) and the fact that $w_1^{\circledast} > 0$ (Lemma 34) imply that

$$J(\mathbf{w}^{\circledast}) \geq 0.69q^2 + 2.4q + 1.7.$$

On the other hand,

$$J(100, 2) \leq -1.5q^2 + 6q + 0.42,$$

and the upper bound on $J(100, 2)$ is less than the lower bound on $J(\mathbf{w}^{\circledast})$ when $0 \leq q \leq 1/2$, giving the desired contradiction. ∎

Now, we're ready to show that $\mathbf{w}^{\circledast}$ correctly classifies $(-\epsilon, 1)$.

Lemma 37 $\epsilon w_1^{\circledast} < w_2^{\circledast}$.

Proof: Let $\mathbf{g}$ be the gradient of J evaluated at $(w_2^{\circledast}/\epsilon, w_2^{\circledast})$. Combining Lemmas 35 and 36, $\mathbf{g} \neq (0, 0)$, so

$$\mathbf{w}^{\circledast} \cdot \mathbf{g} < (w_2^{\circledast}/\epsilon, w_2^{\circledast}) \cdot \mathbf{g}.$$

This implies

$$w_1^{\circledast} \left.\frac{\partial J}{\partial w_1}\right|_{(w_2^{\circledast}/\epsilon, w_2^{\circledast})} < \frac{w_2^{\circledast}}{\epsilon} \left.\frac{\partial J}{\partial w_1}\right|_{(w_2^{\circledast}/\epsilon, w_2^{\circledast})}.$$

Since Lemmas 35 and 36 imply that $g(w_2^{\circledast}/\epsilon, w_2^{\circledast})_1 > 0$, this completes the proof. ∎

Finally, we are ready to work on showing that $(1/10, -1)$ is correctly classified by $\mathbf{w}^{\circledast}$, i.e. that $w_1^{\circledast} > 10 w_2^{\circledast}$.

Lemma 38 *For all $a \in \mathbf{R}$,*

$$\left.\frac{\partial J}{\partial w_1}\right|_{(10a, a)} < 0.$$

Proof: Choose $a \in R$. From (36), we have

$$\begin{aligned}
\left.\frac{\partial J}{\partial w_1}\right|_{(10a,a)} &= q(1-q)\left(\frac{-3}{1+\exp(10a)} + \frac{3\epsilon}{1+\exp(-10\epsilon a)} + \frac{-1}{10(1+\exp(a))}\right) \\
&\quad + (1-q)^2\left(\frac{-3}{1+\exp(10a)} + \frac{3\epsilon}{1+\exp(a-10\epsilon a)} + \frac{-1}{20}\right) \\
&\leq (1-q)^2\left(6\epsilon + \frac{-1}{20}\right) < 0
\end{aligned}$$

using $q \leq 1/2$ and $\epsilon = 1/1000$. ∎

Lemma 39 $w_1^{\circledast} > 10w_2^{\circledast}$.

Proof: Let $\mathbf{g}$ be the gradient of J evaluated at $\mathbf{u} = (10w_2^{\circledast}, w_2^{\circledast})$. Lemma 38 implies that $\mathbf{g} \neq (0,0)$, i.e. that $w_1^{\circledast} \neq 10w_2^{\circledast}$. Therefore,

$$\mathbf{w}^{\circledast} \cdot \mathbf{g} < \mathbf{u} \cdot \mathbf{g}$$

which, since $u_2 = w_2^{\circledast}$, implies

$$w_1^{\circledast} \frac{\partial J}{\partial w_1}\Big|_{\mathbf{u}} < 10w_2^{\circledast} \frac{\partial J}{\partial w_1}\Big|_{\mathbf{u}}.$$

Since Lemma 38 implies that $\partial J/\partial w_1\Big|_{\mathbf{u}} < 0$, this in turn implies

$$w_1^{\circledast} > 10w_2^{\circledast},$$

completing the proof. ∎

Now we have all the pieces to prove that dropout succeeds on P.

Proof (of Theorem 19): Lemma 34 implies that $(1,0)$ is classified correctly by $\mathbf{w}^{\circledast}$, and therefore by $\mathbf{w}^* = p\mathbf{w}^{\circledast}$. Lemma 37 implies that $(-\epsilon, 1)$ is classified correctly. Lemma 39 implies that $(1/10, -1)$ is classified correctly, completing the proof. ∎

Appendix H. Proof of Theorem 20

> **Theorem 20.** If $\lambda \leq \frac{1}{30n}$ then the weight vector $v(P_9, \lambda)$ optimizing the L_2 criterion has perfect prediction accuracy: $\mathrm{er}_{P_9}(v(P_9, \lambda)) = 0$.

In this proof, let us abbreviate P_9 as just P.

By symmetry and convexity, the optimizing $\mathbf{v}$ is of the form $(v_1, v_2, v_2, \ldots, v_2)$ with the last $n-1$ components being equal. Thus for this distribution minimizing the L_2 criterion is equivalent to minimizing the simpler criterion $K(w_1, w_2)$ defined by:

$$K(w_1, w_2) = \frac{9}{10} \ln\left(1 + \exp(-w_1 - w_2)\right) + \frac{1}{10} \ln\left(1 + \exp(w_1 - w_2)\right) + \frac{\lambda}{2}\left(w_1^2 + (n-1)w_2^2\right).$$

Let (v_1, v_2) be the minimizing vector of $K()$, retaining an implicit dependence on n and λ. We will be making frequent use of the partial derivatives of K:

$$\frac{\partial K}{\partial w_1} = \frac{-9}{10(1 + \exp(w_1 + w_2))} + \frac{1}{10(1 + \exp(-w_1 + w_2))} + \lambda w_1 \tag{39}$$

$$\frac{\partial K}{\partial w_2} = \frac{-9}{10(1 + \exp(w_1 + w_2))} + \frac{-1}{10(1 + \exp(-w_1 + w_2))} + (n-1)\lambda w_2. \tag{40}$$

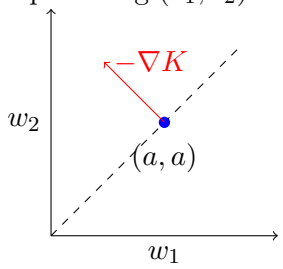
It suffices to show that $0 \leq v_1 < v_2$ so that the first feature does not perturb the majority vote of the others.

To see $0 \leq v_1$, notice that $\partial K/\partial w_1\big|_{(0,w_2)}$ is negative for all w_2, including when $w_2 = v_2$.

To prove $v_1 < v_2$ we show the existence of a point (a, a) such that

$$\left.\frac{\partial K}{\partial w_1}\right|_{(a,a)} = -\left.\frac{\partial K}{\partial w_2}\right|_{(a,a)} > 0, \tag{41}$$

so that Lemma 25 implies that the optimizing (v_1, v_2) lies above the $w_1 = w_2$ diagonal.



We have

$$\left.\frac{\partial K}{\partial w_1}\right|_{(a,a)} = \frac{-9}{10(1+\exp(2a))} + \frac{1}{20} + \lambda a$$

which is increasing in a, negative when $a = 0$ and goes to infinity with a. It turns positive at some $a < 1.5$ (exactly where depends on λ).

On the other hand,

$$\left.\frac{\partial K}{\partial w_2}\right|_{(a,a)} = \frac{-9}{10(1+\exp(2a))} + \frac{-1}{20} + \lambda(n-1)a$$

and is also increasing in a and goes to infinity. However, $\left.\partial K/\partial w_2\right|_{(a,a)}$ is negative at $a = 1.5$ whenever $1.5\lambda(n-1) \leq 1/20$, which is implied by the premise of the theorem.

Both partial derivatives are negative when $a = 0$, continuously go to infinity with a, and $\left.\partial K/\partial w_1\right|_{(a,a)}$ crosses zero first. From the point where $\left.\partial K/\partial w_1\right|_{(a,a)}$ crosses zero until $\left.\partial K/\partial w_2\right|_{(a,a)}$ does, the magnitude of $\left.\partial K/\partial w_1\right|_{(a,a)}$ is increasing, starting at 0, and the magnitude of $\left.\partial K/\partial w_2\right|_{(a,a)}$ is decreasing until it reaches 0. When they meet, Equation (41) holds, completing the proof.

Appendix I. Proof of Theorem 21

> **Theorem 21.** If the dropout probability $q = 1/2$ and the number of features is an even $n > 125$ then the weight vector $\mathbf{w}^*(P_9, q)$ optimizing the dropout criterion has prediction error rate $\mathrm{er}_{P_9}(\mathbf{w}^*(P_9, q)) \geq 1/10$.

In this proof, we again abbreviate, using P for P_9.

The complicated form of the criterion optimized by dropout makes analyzing it difficult. Here we make use of Jensen's inequality. However, a straightforward application of it is fruitless, and a key step is to apply Jensen's inequality on just half the distribution resulting from dropout.

Similarly to before, let

$$J(\mathbf{w}) = \mathbf{E}_{(\mathbf{x},y)\sim P,\mathbf{r}}(\ell(y(\mathbf{w}\cdot(\mathbf{r}\odot\mathbf{x})))), \tag{42}$$

and let $\mathbf{w}^{\circledast}$ minimize J, so that $\mathbf{w}^* = p\mathbf{w}^{\circledast}$.

Again using symmetry and convexity, the last $n-1$ components of the optimizing $\mathbf{w}^{\circledast}$ are equal, so $\mathbf{w}^{\circledast}$ is of the form $(w_1^{\circledast}, w_2^{\circledast}, w_2^{\circledast}, \ldots, w_2^{\circledast})$.

Lemma 40 *The minimizing $w_1^{\circledast}$ of (42) is positive.*

Proof: Let $\widetilde{P,\mathbf{r}}$ be the marginal distribution of the last $n-1$ components after dropout and $\tilde{\mathbf{x}}$ denote these last $n-1$ components of the dropped-out feature vector. Then, recalling y is always 1 in our distribution (and p is the probability that the first feature is *not* dropped out),

$$\frac{\partial J(w)}{\partial w_1} = \mathbf{E}_{(r_2,\ldots,r_n)}\left(\frac{9p}{10}\mathbf{E}_{\tilde{\mathbf{x}}\sim\widetilde{P,\mathbf{r}}}(\ell'(\mathbf{w}\cdot(1,\tilde{\mathbf{x}}))) - \frac{p}{10}\mathbf{E}_{\tilde{\mathbf{x}}\sim\widetilde{P,\mathbf{r}}}(\ell'(\mathbf{w}\cdot(-1,\tilde{\mathbf{x}})))\right)$$

which is negative whenever $w_1 = 0$, since $\ell'()$ is negative and the two inner expectations become identical when $w_1 = 0$. Therefore the optimizing $w_1^{\circledast}$ is positive. ∎

To show that dropout fails, we want to show that $w_1^{\circledast} > w_2^{\circledast}$, i.e. that $w_1^{\circledast} \le w_2^{\circledast}$ leads to a contradiction, so we begin to explore the consequences of $w_1^{\circledast} \le w_2^{\circledast}$.

Lemma 41 *If $q = 1/2$ and $w_1^{\circledast} \le w_2^{\circledast}$ then $w_2^{\circledast} > 4/9$.*

Proof: Assume to the contrary that $w_1^{\circledast} \le w_2^{\circledast} \le 4/9$.

Using Jensen's inequality,

$$J(\mathbf{w}^{\circledast}) \ge \ell(\mathbf{E}_{(\mathbf{x},y)\sim P,\mathbf{r}}(y(\mathbf{w}^{\circledast}\cdot\mathbf{x})))$$

and the inner expectation is $8w_1^{\circledast}/20 + w_2^{\circledast}/2 \le 9w_2^{\circledast}/10$ as $w_1^{\circledast} \le w_2^{\circledast}$. Therefore, since $w_2^{\circledast} \le 4/9$,

$$J(\mathbf{w}^{\circledast}) \ge \ell(0.4) > 0.51.$$

However,

$$J(2.1, 0, 0, \ldots, 0) = \frac{\ln(2)}{2} + \frac{9\ln(1+e^{-2.1})}{20} + \frac{\ln(1+e^{2.1})}{20} < 0.51$$

contradicting the optimality of $\mathbf{w}^{\circledast}$. ∎

Lemma 42 *If $q = 1/2$ and $w_1^{\circledast} \le w_2^{\circledast}$ then $J(\mathbf{w}^{\circledast}) \ge \mathbf{E}_{k\sim B(n,1/2)}\ell(w_2^{\circledast}(k-(n/2)+1))$ where $B(n,1/2)$ is the binomial distribution.*

Proof: Consider the modified distribution P_1 over $(\mathbf{x}, y)$ examples where y is always 1, x_2, ..., x_n are uniformly distributed over the the vectors with $n/2$ ones and $(n/2)-1$ negative

ones (as in P), but x_1 is always one. Since $0 < w_1^{\circledast} \leq w_2^{\circledast}$ and the label $y = 1$ under P and P_1,

$$\begin{aligned} J(\mathbf{w}^{\circledast}) &= \mathbf{E}_{(\mathbf{x},y)\sim P,\mathbf{r}}(\ell(\mathbf{w}^{\circledast} \cdot \mathbf{x})) \\ &> \mathbf{E}_{(\mathbf{x},y)\sim P_1,\mathbf{r}}(\ell(\mathbf{w}^{\circledast} \cdot \mathbf{x})) \\ &= \mathbf{E}_{(\mathbf{x},y)\sim P_1,\mathbf{r}}\left(\ell\big(w_2^{\circledast}(\mathbf{1} \cdot (\mathbf{x} \odot \mathbf{r}))\big)\right) \\ &= \mathbf{E}_{(\mathbf{x},y)\sim P_1,\mathbf{r}}\left(\ell\big(w_2^{\circledast}(\mathbf{x} \cdot \mathbf{r})\big)\right). \end{aligned}$$

Every $\mathbf{x}$ in the support of P_1 has exactly $(n/2)+1$ components that are 1, and the remaining $(n/2)-1$ components are -1. Call a component a *success* if it is either -1 and dropped out or 1 and not dropped out. Now, $\mathbf{x} \cdot \mathbf{r}$ is exactly $1-(n/2)$ plus the number of successes. Furthermore, the number of successes is distributed according to the binomial distribution $B(n, 1/2)$. Therefore

$$\mathbf{E}_{(\mathbf{x},y)\sim P_1,\mathbf{r}}(w_2^{\circledast}(\mathbf{x} \cdot \mathbf{r})) = \mathbf{E}_{k\sim B(n,1/2)}(\ell(w_2^{\circledast}(k-(n/2)+1)))$$

giving the desired bound. ∎

Lemma 43 *For even $n \geq 6$, $\mathbf{E}_{k\sim B(n,1/2)}(\ell(w_2^{\circledast}(k-(n/2)+1))) \geq \frac{1}{3}\ell\left(w_2^{\circledast} - \frac{w_2^{\circledast}\sqrt{2n}}{4}\right)$.*

Proof: Let $\alpha = \sum_{i=0}^{n/2-1}\binom{n}{i}$, so α is slightly less than 2^{n-1}.

$$\begin{aligned} \mathbf{E}_{k\sim B(n,1/2)}(\ell(w_2^{\circledast}(k-(n/2)+1))) &= \frac{1}{2^n}\sum_{k}\binom{n}{k}\ell(w_2^{\circledast}(k+1-(n/2))) \\ &> \frac{\alpha}{2^n}\sum_{k=0}^{n/2-1}\frac{1}{\alpha}\binom{n}{k}\ell(w_2^{\circledast}(1+k-(n/2))) \\ &> \frac{\alpha}{2^n}\ell\left(\sum_{k=0}^{n/2-1}\frac{1}{\alpha}\binom{n}{k}w_2^{\circledast}(1+k-(n/2))\right) \end{aligned}$$

where the last step uses Jensen's inequality. Continuing,

$$\mathbf{E}_{k\sim B(n,1/2)}(\ell(w_2^{\circledast}(k-(n/2)+1))) > \frac{\alpha}{2^n}\ell\left(w_2^{\circledast} + \frac{w_2^{\circledast}}{\alpha}\sum_{k=0}^{n/2-1}\binom{n}{k}(k-(n/2))\right).$$

Equation (5.18) of Concrete Mathematics (Graham et al., 1989) and the bound $\binom{n}{n/2} \geq \frac{2^n}{\sqrt{2n}}$ give

$$\sum_{k=0}^{n/2-1}\binom{n}{k}(k-(n/2)) = \frac{-n}{4}\binom{n}{n/2} \leq \frac{-\sqrt{2n}\,2^{n-1}}{4}.$$

Therefore, recalling that $\alpha < 2^{n-1}$ and noting $\alpha/2^n > 1/3$ when $n \geq 6$,

$$\begin{aligned}\mathbf{E}_{k\sim B(n,1/2)}(\ell(w_2^{\circledast}(k-(n/2)+1))) &> \frac{\alpha}{2^n}\ell\left(w_2^{\circledast} - \frac{w_2^{\circledast}}{\alpha}\frac{2^{n-1}\sqrt{2n}}{4}\right) \\ &> \frac{1}{3}\ell\left(w_2^{\circledast} - \frac{w_2^{\circledast}\sqrt{2n}}{4}\right).\end{aligned}$$

∎

We now have the necessary tools to prove Theorem 21.

Proof: (of Theorem 21) If $w_1^{\circledast} > w_2^{\circledast}$ then the first feature will dominate the majority vote of the others and the optimizing $\mathbf{w}^{\circledast}$ has prediction error rate 1/10 . We now assume to the contrary that $w_1^{\circledast} \leq w_2^{\circledast}$. When $n > 125$ and $w_2^{\circledast} \geq 4/9$ (from Lemma 41) we have

$$w_2^{\circledast} - \frac{w_2^{\circledast}\sqrt{2n}}{4} \leq -1.31$$

and $\ell(w_2^{\circledast} - \frac{w_2^{\circledast}\sqrt{2n}}{4}) > 1.54$.

Lemmas 42 and 43 now imply that $J(\mathbf{w}^{\circledast}) > 0.51$, but (as in Lemma 41) $J(2.1, 0, \ldots 0) < 0.51$, contradicting the optimality of $\mathbf{w}^{\circledast}$. ∎

Many of the approximations used to prove Theorem 21 are quite loose, resulting in large values of n being needed to obtain the contradiction. For this class of distributions and $q = 1/2$ we conjecture that optimizing the dropout criterion fails to produce the Bayes optimal hypothesis for every even $n \geq 4$.

Appendix J. Proof of Theorem 22

Theorem 22. If dropout probability $q = 1/2$ and the number of features is $n = 4$ then the minimizer of the dropout criteria $\mathbf{w}^*(P_9, q)$ has has prediction error rate $\mathrm{er}_{P_9}(\mathbf{w}^*(P_9, q)) \geq 1/10$.

In this proof, let us also refer to P_9 as just P and let $\mathbf{w}^{\circledast}$ be the minimizer of (42).

As before, the optimizing $\mathbf{w}^{\circledast}$ has the form $(w_1^{\circledast}, w_2^{\circledast}, w_2^{\circledast}, w_2^{\circledast})$ by symmetry and convexity. Recalling that the label y is always 1 under distribution P, we can use the equivalent criterion

$$K(w_1, w_2) = \mathbf{E}_{(\mathbf{x},y)\sim P,\mathbf{r}}(\ell(y(\mathbf{w}\cdot\mathbf{x}))) = \mathbf{E}_{(\mathbf{x},y)\sim P,\mathbf{r}}\left(\ell\left(w_1x_1r_1 + w_2\sum_{i=2}^{4} x_ir_i\right)\right).$$

This expectation can be written with 12 terms, one for each pairing of the three possible x_1r_1 values with the four possible $\sum_{i=2}^{4} x_ir_i \in \{-1, 0, 1, 2\}$ values (see Table 3).

Taking them in order, we have

$$\begin{aligned}K(w_1, w_2) =& \frac{9}{160}\ell(w_1 + 2w_2) + \frac{27}{160}\ell(w_1 + w_2) + \frac{27}{160}\ell(w_1) + \frac{9}{160}\ell(w_1 - w_2) \\ &+ \frac{10}{160}\ell(2w_2) + \frac{30}{160}\ell(w_2) + \frac{30}{160}\ell(0) + \frac{10}{160}\ell(w_2) \\ &+ \frac{1}{160}\ell(-w_1 + 2w_2) + \frac{3}{160}\ell(-w_1 + w_2) + \frac{3}{160}\ell(-w_1) + \frac{1}{160}\ell(-w_1 - w_2).\end{aligned}$$

$x_1 r_1$	probability	$\sum_{i=2}^{4} x_i r_i$	probability
1	9/20	2	1/8
0	1/2	1	3/8
-1	1/20	0	3/8
		-1	1/8

Table 3: Probabilities of $x_1 r_1$ and $\sum_{i=2}^{4} x_i r_i$ values assuming dropout probability $q = 1/2$.

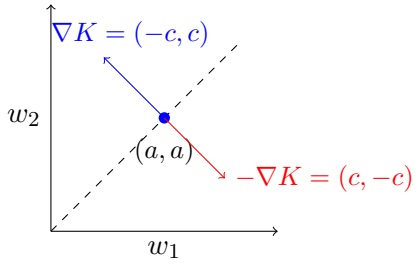


Figure 8: If ∇K at some (a, a) is $(-c, c)$ for some $c > 0$ then $w_1^{\circledast} > w_2^{\circledast}$.

So, when $p = q = 1/2$, the derivatives are:

$$\frac{\partial K}{\partial w_1}$$
$$= \frac{1}{160}\left(\frac{-9}{1+\exp(w_1+2w_2)} + \frac{-27}{1+\exp(w_1+w_2)} + \frac{-27}{1+\exp(w_1)} + \frac{-9}{1+\exp(w_1-w_2)}\right.$$
$$\left. + \frac{1}{1+\exp(-w_1+2w_2)} + \frac{3}{1+\exp(-w_1+w_2)} + \frac{3}{1+\exp(-w_1)} + \frac{1}{1+\exp(-w_1-w_2)}\right),$$
$$\frac{\partial K}{\partial w_2}$$
$$= \frac{1}{160}\left(\frac{-18}{1+\exp(w_1+2w_2)} + \frac{-27}{1+\exp(w_1+w_2)} + \frac{9}{1+\exp(w_1-w_2)}\right.$$
$$+ \frac{-20}{1+\exp(2w_2)} + \frac{-30}{1+\exp(w_2)} + \frac{10}{1+\exp(-w_2)}$$
$$\left. + \frac{-2}{1+\exp(-w_1+2w_2)} + \frac{-3}{1+\exp(-w_1+w_2)} + \frac{1}{1+\exp(-w_1-w_2)}\right).$$

If $w_1^{\circledast} > w_2^{\circledast}$, then dropout will have prediction error rate 1/10 as $w_1^{\circledast}$ will dominate the vote of the other three components. We show that $w_1^{\circledast} > w_2^{\circledast}$ by proving that there is a point (a, a) in weight space such that the gradient at (a, a) is of the form $(-c, c)$ for some $c > 0$ (see Figure 8).

The derivatives when evaluated at (a,a) are:

$$\left.\frac{\partial K}{\partial w_1}\right|_{(a,a)} = \frac{1}{160}\left(\frac{-9}{1+\exp(3a)} + \frac{-27}{1+\exp(2a)} + \frac{-26}{1+\exp(a)} - 3 + \frac{3}{1+\exp(-a)} + \frac{1}{1+\exp(-2a)}\right)$$

and

$$\left.\frac{\partial K}{\partial w_2}\right|_{(a,a)} = \frac{1}{160}\left(\frac{-18}{1+\exp(3a)} + \frac{-47}{1+\exp(2a)} + \frac{-32}{1+\exp(a)} + 3 + \frac{10}{1+\exp(-a)} + \frac{1}{1+\exp(-2a)}\right).$$

Note that both of these derivatives are increasing in a, positive for large a, and negative when $a = 0$. At $a = 2\ln(2)$, derivative $\partial K/\partial w_1\big|_{(a,a)}$ is still negative, while $\partial K/\partial w_1\big|_{(a,a)}$ has turned positive, so $\partial K/\partial w_1\big|_{(a,a)}$ crosses 0 first. The continuity of the partial derivatives now implies the existence of an (a,a) where ∇K has the form $(-c,c)$, completing the proof. ∎

Appendix K. Proof of Theorem 23

Theorem 23. If $q = 1/2$, $n \geq 100$, $\alpha > 0$, $\beta = 1/(10\sqrt{n-1})$, and $\eta \leq \frac{1}{2+\exp(54\sqrt{n})}$, then $\mathrm{er}_{P_{10}}(\mathbf{w}^*(P_{10}, q)) = \eta$.

For this subsection, let $P = P_{10}$ and define the scaled dropout criterion

$$J(\mathbf{w}) = \mathbf{E}_{(\mathbf{x},y)\sim P,\mathbf{r}}(\ell(y\mathbf{w}\cdot(\mathbf{r}\odot\mathbf{x}))),$$

where, as earlier, the components of $\mathbf{r}$ are independent samples from a Bernoulli distribution with parameter $p = 1-q = 1/2 > 0$. Let $\mathbf{w}^{\circledast}$ be the minimizer of J, so that $\mathbf{w}^* = p\mathbf{w}^{\circledast}$.

Note that, by symmetry, the contribution to J from the cases where y is -1 and 1 respectively are the same, so the value of J is not affected if we clamp y at 1. Let us use this form to express J, and let D be the marginal distribution of feature vector $\mathbf{x}$ conditioned on the label $y = 1$.

Let $B = \{2, ..., n\}$. By symmetry, $w_i^{\circledast}$ is identical for all $i \in B$ so $\mathbf{w}^{\circledast}$ is the minimum of J over weight vectors satisfying this constraint. Let $K(w_1, w_2) = J(w_1, w_2, ..., w_2)$; note that $w_1^{\circledast}, w_2^{\circledast}$ minimizes K defined by

$$K(w_1, w_2) = \mathbf{E}_{\mathbf{x}\sim D,\mathbf{r}}(\ell(w_1 r_1 x_1 + w_2 \sum_{i\in B} r_i x_i)).$$

To prove Theorem 23, it suffices to show that

$$w_1^{\circledast} > (n-1)w_2^{\circledast}/\alpha > 0, \tag{43}$$

since when (43) holds, $\mathbf{w}^{\circledast}$ always outputs x_1.

We have

$$\frac{\partial K}{\partial w_1} = \frac{1}{2}\mathbf{E}_{\mathbf{x}\sim D,\mathbf{r}}\left(\frac{-x_1}{1+\exp(w_1x_1 + w_2\sum_{i\in B} r_i x_i)}\right) \tag{44}$$

$$\frac{\partial K}{\partial w_2} = \mathbf{E}_{\mathbf{x}\sim D,\mathbf{r}}\left(\frac{-\sum_{i\in B} r_i x_i}{1+\exp(w_1r_1x_1 + w_2\sum_{i\in B} r_i x_i)}\right). \tag{45}$$

(Note that, in (44), we have marginalized out r_1.)

Lemma 44 $w_2^{\circledast} > 0$.

As before, it suffices to show that there is a point $(a_1, 0)$ where both $\frac{\partial K}{\partial w_2}\big|_{(a_1,0)} < 0$ and $\frac{\partial K}{\partial w_1}\big|_{(a_1,0)} = 0$. From equation (45),

$$\frac{\partial K}{\partial w_2}\big|_{(a_1,0)} = \mathbf{E}_{\mathbf{x}\sim D,\mathbf{r}}\left(\frac{-\sum_{i\in B} r_i x_i}{1+\exp(a_1r_1x_1)}\right) < 0$$

for all real a_1.

Now, evaluating (44), dividing into cases based on x_1, we get

$$\frac{\partial K}{\partial w_1}\big|_{(a_1,0)} = (\eta/2)\left(\frac{\alpha}{1+\exp(-\alpha a_1)}\right) + ((1-\eta)/2)\left(\frac{-\alpha}{1+\exp(\alpha a_1)}\right).$$

This approaches $-\alpha((1-\eta)/2)$ as a_1 approaches $-\infty$, and it approaches $\alpha\eta/2$ as a_1 approaches ∞. Since it is a continuous function of a_1, there must be a value of a_1 such that $\frac{\partial K}{\partial w_1}\big|_{(a_1,0)} = 0$. Putting this together with $\frac{\partial K}{\partial w_2}\big|_{(a_1,0)} < 0$ completes the proof. ∎

To show the sufficient inequalities (43), it will be useful to prove an upper bound on $w_2^{\circledast}$. (This upper bound will make it easier to show, informally, that $w_1^{\circledast}$ is needed.) In order to bound the size of $w_2^{\circledast}$, we will prove a lower bound on K in terms of w_2. For this, we want to show that, if w_2 is too large, then the algorithm will pay too much when it makes large-margin errors. For *this*, we need a lower bound on the probability of a large-margin error. For this, we can adapt an analysis that provided a lower bound on the probability of an error from (Helmbold and Long, 2012).

To simplify the proof, we will first provide a lower bound on the dropout risk in terms of the risk without dropout. We will actually prove something somewhat more general, for possible future reference.

Lemma 45 *Let* $\mathbf{r}$ *and* $\mathbf{x}$ *be independent,* $\mathbf{R}^N$*-valued random variables; let* ϕ *be convex function of a scalar real variable. Then*

$$\mathbf{E}_{\mathbf{r},\mathbf{x}}\left(\phi\left(\sum_i x_i r_i\right)\right) \geq \mathbf{E}_{\mathbf{x}}\left(\phi\left(\sum_i x_i \mathbf{E}_{\mathbf{r}}(r_i)\right)\right).$$

Proof: Since $\mathbf{x}$ and $\mathbf{r}$ are independent,

$$\begin{aligned}
&\mathbf{E}_{\mathbf{r},\mathbf{x}}(\phi(\sum_i x_i r_i)) \\
&= \mathbf{E}_{\mathbf{x}}(\mathbf{E}_{\mathbf{r}}(\phi(\sum_i x_i r_i))) \\
&\geq \mathbf{E}_{\mathbf{x}}(\phi(\mathbf{E}_{\mathbf{r}}(\sum_i x_i r_i))) \quad \text{(by Jensen's Inequality)} \\
&= \mathbf{E}_{\mathbf{x}}(\phi(\sum_i x_i \mathbf{E}_{\mathbf{r}}(r_i))),
\end{aligned}$$

completing the proof. ∎

Now, it is enough to lower bound the probability of a large-margin error with respect to the original distribution. Recall $B = \{2, \ldots, n\}$.

Lemma 46 $\mathbf{Pr}\left(\frac{1}{n-1}\sum_{i\in B} x_i < -2\beta\right) \geq \frac{3}{10}$.

Proof: If Z is a standard normal random variable and R is a binomial (ℓ, p) random variable with $p \leq 1/2$, then for $\ell(1-p) \leq j \leq \ell p$, Slud's inequality (Slud, 1977) gives

$$\mathbf{Pr}(R \geq j) \geq \mathbf{Pr}\left(Z \geq \frac{j - \ell p}{\sqrt{\ell p(1-p)}}\right), \tag{46}$$

as worked out in Lemma 23 of (Helmbold and Long, 2012).

Now, we have

$$\begin{aligned}
\mathbf{Pr}\left(\frac{1}{n-1}\sum_{i\in B} x_i < -2\beta\right) &= \mathbf{Pr}\left(\sum_{i\in B} x_i/2 < -(n-1)\beta\right) \\
&= \mathbf{Pr}\left(\sum_{i\in B} (x_i+1)/2 < (n-1)/2 - (n-1)\beta\right) \\
&= \mathbf{Pr}\left(\sum_{i\in B} z_i < (n-1)(1/2-\beta)\right)
\end{aligned}$$

where the z_i's are independent $\{0,1\}$-valued variables with $\mathbf{Pr}(z_i = 1) = 1/2 + \beta$. Let $\bar{z}_i$ be $1 - z_i$, so $\sum_{i\in B} \bar{z}_i$ is a Binomial $(n-1, 1/2-\beta)$ random variable. Furthermore,

$$\begin{aligned}
\mathbf{Pr}\left(\sum_{i\in B} z_i < (n-1)(1/2-\beta)\right) &= \mathbf{Pr}\left(\sum_{i\in B} \bar{z}_i > (n-1) - (n-1)(1/2-\beta)\right) \\
&= \mathbf{Pr}\left(\sum_{i\in B} \bar{z}_i > (n-1)(1/2+\beta)\right).
\end{aligned}$$

Using (46) with $j = (n-1)(1/2+\beta)$, $\ell = (n-1)$, and $p = 1/2-\beta$ gives:

$$\begin{aligned}\mathbf{Pr}\left(\sum_{i\in B}\bar{z}_i > (n-1)(1/2+\beta)\right) &\geq \mathbf{Pr}\left(Z \geq \frac{(n-1)(1/2+\beta)-(n-1)(1/2-\beta)}{\sqrt{(n-1)(1/4-\beta^2)}}\right)\\ &= \mathbf{Pr}\left(Z \geq \frac{2(n-1)\beta}{\sqrt{(n-1)(1/4-\beta^2)}}\right).\end{aligned}$$

Since $\beta = 1/(10\sqrt{n})$ and $n \geq 100$, this implies

$$\mathbf{Pr}\left(\frac{1}{n-1}\sum_{i\in B}x_i < -2\beta\right) \geq \mathbf{Pr}\left(Z \geq 1/2\right).$$

Since the density of Z is always at most $1/\sqrt{2\pi}$, we have

$$\mathbf{Pr}\left(\frac{1}{n-1}\sum_{i\in B}x_i < -2\beta\right) \geq \mathbf{Pr}(Z\geq 0) - \mathbf{Pr}(Z\in(0,1/2)) > \frac{1}{2} - \frac{1}{2\sqrt{2\pi}} > 3/10,$$

completing the proof. ∎

Now we are ready for the lower bound on the dropout risk in terms of w_2.

Lemma 47 *For all w_1,*

$$K(w_1,w_2) > \frac{w_2\sqrt{n-1}}{67}.$$

Proof: Considering only the case in which x_1 is dropped out (i.e. $r_1 = 0$), we have

$$K(w_1,w_2) \geq \frac{1}{2}\mathbf{E}\left(\ell\left(w_2\sum_i r_i x_i\right)\right).$$

Applying Lemma 45, we get

$$K(w_1,w_2) \geq \frac{1}{2}\mathbf{E}\left(\ell\left((w_2/2)\sum_{i\in B}x_i\right)\right).$$

Since ℓ is non-increasing and non-negative, we have

$$K(w_1,w_2) \geq \frac{1}{2}\ell(-w_2\beta(n-1))\mathbf{Pr}\left(\frac{1}{n-1}\sum_{i\in B}x_i < -2\beta\right),$$

and applying Lemma 46 gives

$$K(w_1,w_2) \geq \frac{3\ell(-w_2\beta(n-1))}{20}.$$

Since $\ell(z) > -z$, we have

$$K(w_1,w_2) \geq \frac{3w_2\beta(n-1)}{20}$$

and, using $\beta = \frac{1}{10\sqrt{n-1}}$, we get

$$K(w_1, w_2) \geq \frac{3w_2\sqrt{n-1}}{200},$$

completing the proof. ∎

Lemma 48 $w_2^{\circledast} < \frac{27}{\sqrt{n-1}}$.

Proof: Note that

$$K(w, 0) = \ell(0)/2 + (1/2)(\eta\ell(-\alpha w) + (1-\eta)\ell(\alpha w)),$$

is increasing in η so that

$$K(w_1^{\circledast}, w_2^{\circledast}) \leq K(5/\alpha, 0) < \ell(0)/2 + 1/35 \tag{47}$$

since $\eta < 1/100$.

On the other hand, Lemma 47 gives

$$K(w_1^{\circledast}, w_2^{\circledast}) > \frac{w_2\sqrt{n-1}}{67}.$$

Solving for $w_2^{\circledast}$ completes the proof. ∎

Lemma 49 *For all* $0 < u < \frac{27}{\sqrt{n-1}}$, *we have*

$$\frac{\partial K}{\partial w_1}\Big|_{((n-1)u/\alpha, u)} < 0.$$

Proof: From (44), we have

$$\begin{aligned}
&2\frac{\partial K}{\partial w_1}\Big|_{(nu/\alpha, u)} \\
&= \mathbf{E}_{\mathbf{x}\sim D, \mathbf{r}}\left(\frac{-x_1}{1+\exp((n-1)ux_1/\alpha + u\sum_{i\in B} r_i x_i)}\right) \\
&= \eta\mathbf{E}_{\mathbf{x}\sim D, \mathbf{r}}\left(\frac{\alpha}{1+\exp(-(n-1)u + u\sum_{i\in B} r_i x_i)}\right) \\
&\quad + (1-\eta)\mathbf{E}_{\mathbf{x}\sim D, \mathbf{r}}\left(\frac{-\alpha}{1+\exp((n-1)u + u\sum_{i\in B} r_i x_i)}\right) \\
&< \eta\alpha + (1-\eta)\mathbf{E}_{\mathbf{x}\sim D, \mathbf{r}}\left(\frac{-\alpha}{1+\exp((n-1)u + u\sum_{i\in B} r_i x_i)}\right) \\
&< \alpha\left(\eta + \frac{-(1-\eta)}{1+\exp(2(n-1)u)}\right) \quad (\text{since } \textstyle\sum_{i\in B} r_i x_i \leq n-1) \\
&< \alpha\left(\eta + \frac{-(1-\eta)}{1+\exp(54\sqrt{n-1}}\right) \quad (\text{since } u < 27/\sqrt{n-1}) \\
&< 0
\end{aligned}$$

since $\eta \leq 1/(2+\exp(54\sqrt{n}))$, completing the proof. ∎

Recall that, to prove Theorem 23, since we already showed $w_2^{\circledast} > 0$, all we needed was to show that $\alpha w_1^{\circledast} > (n-1)w_2^{\circledast}$. We do this next.

Lemma 50 $\alpha w_1^{\circledast} > (n-1)w_2^{\circledast}$.

Proof: Let $\mathbf{g}$ be the gradient of J evaluated at $\mathbf{u} = ((n-1)w_2^{\circledast}/\alpha, w_2^{\circledast})$. Lemmas 48 and 49 implies that $\mathbf{g} \neq (0,0)$. By convexity

$$\mathbf{w}^{\circledast} \cdot \mathbf{g} < \mathbf{u} \cdot \mathbf{g}$$

which, since $u_2 = w_2^{\circledast}$, implies

$$w_1^{\circledast} g_1 < (n-1)w_2^{\circledast} g_1/\alpha.$$

Since, by Lemmas 48 and 49, $g_1 < 0$,

$$w_1^{\circledast} > (n-1)w_2^{\circledast}/\alpha$$

completing the proof. ∎

Appendix L. Proof of Theorem 24

Theorem 24. If $\beta = 1/(10\sqrt{n-1})$, $\lambda = \frac{1}{30n}$, $\alpha < \beta\lambda$, and n is a large enough even number, then for any $\eta \in [0,1]$, $\mathrm{er}_{P_{10}}(\mathbf{v}(P_{10}, \lambda)) \geq 3/10$.

In this proof, let us also abbreviate P_{10} with P and use J to denote the L_2 regularized criterion in Equation (5) specialized for the distribution of this P.

As before, the contribution to the L_2 criterion from the cases where y is -1 and 1 respectively are the same, so the value of the criterion is not affected if we clamp y at 1. Furthermore, we leave the dependency on λ implicit and (since the source is fixed) use the more succinct $\mathbf{v}$ for $\mathbf{v}(P, \lambda)$.

Also, if, as before, we let $B = \{2, ..., n\}$, then by symmetry, v_i is identical for all $i \in B$ so $\mathbf{v}$ is the minimum of J over weight vectors satisfying this constraint. Let $K(w_1, w_2) = J(w_1, w_2, ..., w_2)$ so that (v_1, v_2) minimizes K. Recall that D is the marginal distribution of $\mathbf{x}$ under P conditioned on $y = 1$.

$$K(w_1, w_2) = \mathbf{E}_{\mathbf{x}\sim D}\left(\ell\left(w_1 x_1 + w_2 \sum_{i\in B} x_i\right)\right) + \frac{\lambda}{2}(w_1^2 + (n-1)w_2^2).$$

Lemma 46, together with the fact that $|x_1| = \alpha$, implies that,

$$\alpha v_1 < 2\beta(n-1)v_2 \tag{48}$$

suffices to prove Theorem 24, so we set this as our subtask.

We have

$$\frac{\partial K}{\partial w_1} = \mathbf{E}_{\mathbf{x}\sim D}\left(\frac{-x_1}{1 + \exp(w_1 x_1 + w_2 \sum_{i\in B} x_i)}\right) + \lambda w_1 \tag{49}$$

$$\frac{\partial K}{\partial w_2} = \mathbf{E}_{\mathbf{x}\sim D}\left(\frac{-\sum_{i\in B} x_i}{1 + \exp(w_1 x_1 + w_2 \sum_{i\in B} x_i)}\right) + \lambda(n-1)w_2. \tag{50}$$

First, we need a rough bound on v_1.

Lemma 51 $|v_1| \leq \frac{\alpha}{\lambda} < \beta$.

Proof: The second inequality follows from the constraint on α. From (49), we get

$$|v_1| \leq \frac{1}{\lambda}\mathbf{E}_{\mathbf{x}\sim D}\left(\left|\frac{x_1}{1+\exp(v_1x_1+v_2\sum_{i\in B}x_i)}\right|\right)$$

and the facts $|x_1| \leq \alpha$ and $0 < \frac{1}{1+\exp(v_1x_1+v_2\sum_{i\in B}x_i)} \leq 1$ then imply $|v_1| \leq \alpha/\lambda$. ∎

Lemma 52 *For large enough n,*

$$\mathbf{Pr}\left(\sum_{i\in B} x_i \in [\beta(n-1), 3\beta(n-1)]\right) \geq \frac{1}{13}.$$

Proof: Let $\Phi(z) = \mathbf{Pr}(Z \leq z)$ for a standard normal random variable Z and let $S = \sum_{i\in B} x_i$. Note that $\mathbf{E}(x_i) = 2\beta$, $\mathbf{var}(x_i) = 1-4\beta^2$, and the third moment $\mathbf{E}(|x_i-\mathbf{E}(x_i)|^3) = 1-16\beta^4$. The Berry-Esseen inequality (DasGupta, 2008, Theorem 11.1) relates binomial distributions to the normal distribution using these moments, and directly implies that

$$\sup_z \left|\mathbf{Pr}\left(\frac{S}{n-1} - 2\beta \leq \sqrt{\frac{1-4\beta^2}{n-1}} \times z\right) - \Phi(z)\right| \leq \frac{C(1-16\beta^4)}{(1-4\beta^2)^{3/2}\sqrt{n-1}} < \frac{1}{\sqrt{n-1}}$$

where the last inequality follows from the facts that the Berry-Esseen global constant $C \leq 0.8$ and $\beta < 1/10$.

Using the change of variable $s = \sqrt{(1-4\beta^2)(n-1)}\, z + 2\beta(n-1)$ this can be restated:

$$\sup_s \left|\mathbf{Pr}\left(S \leq s\right) - \Phi\left(\frac{s-2\beta(n-1)}{\sqrt{(1-4\beta^2)(n-1)}}\right)\right| \leq \frac{1}{\sqrt{n-1}},$$

so

$$\begin{aligned}
&\mathbf{Pr}(S \in [\beta(n-1), 3\beta(n-1)]) \\
&\geq \mathbf{Pr}_{z\in N(0,1)}\left(z \in \left[-\beta\sqrt{\frac{n-1}{1-4\beta^2}}, \beta\sqrt{\frac{n-1}{1-4\beta^2}}\right]\right) - \frac{2}{\sqrt{n-1}} \\
&\geq \mathbf{Pr}_{z\in N(0,1)}\left(z \in \left[\frac{-1}{10}, \frac{1}{10}\right]\right) - \frac{2}{\sqrt{n-1}} \\
&\geq \frac{1}{13},
\end{aligned}$$

for large enough n. ∎

Recent work shows that the Berry-Esseen constant C is less then $1/2$, this allows us to replace the $2\sqrt{n-1}$ with $1/\sqrt{n-1}$, but it still requires n on the order of 150,000 to get the $1/13$ bound. Reducing the bound to $1/50$ would make n as small as 300 sufficient.

Next, we need a rough bound on v_2.

Lemma 53 $v_2 \geq \frac{1}{n-1}$.

Proof: From (50), we have

$$v_2 = \frac{1}{\lambda(n-1)} \mathbf{E}_{\mathbf{x}\sim D}\left(\frac{\sum_{i\in B} x_i}{1+\exp(v_1 x_1 + v_2 \sum_{i\in B} x_i)}\right).$$

If we denote $\sum_{i\in B} x_i$ by S, then

$$v_2 = \frac{1}{\lambda(n-1)} \mathbf{E}_{\mathbf{x}\sim D}\left(\frac{S}{1+\exp(v_1 x_1 + v_2 S)}\right).$$

Since, for all odd[6] $s > 0$

$$\frac{\mathbf{Pr}(S=s)}{\mathbf{Pr}(S=-s)} = \left(\frac{1+2\beta}{1-2\beta}\right)^s$$

so $\mathbf{Pr}(S=-s) = \mathbf{Pr}(S=s)\left(\frac{1-2\beta}{1+2\beta}\right)^s$. Analyzing the contributions of s and $-s$ together we have

$$\begin{aligned}
&v_2\lambda(n-1)\\
&= \sum_{s=1}^{n-1} \mathbf{Pr}(S=s)\Big((1-\eta)\frac{s}{1+\exp(v_1\alpha + v_2 s)} + \eta\frac{s}{1+\exp(-v_1\alpha + v_2 s)}\\
&+ \left((1-\eta)\frac{-s}{1+\exp(v_1\alpha - v_2 s)} + \eta\frac{-s}{1+\exp(-v_1\alpha - v_2 s)}\right)\left(\frac{1-2\beta}{1+2\beta}\right)^s\Big).
\end{aligned}$$

Recalling that $|v_1| \le \alpha/\lambda$ (Lemma 51), and using the minimizing value in this range for each term gives

$$\begin{aligned}
&v_2\lambda(n-1)\\
&\ge \sum_{s=1}^{n-1} \mathbf{Pr}(S=s)\left(\frac{s}{1+\exp(\alpha^2/\lambda + v_2 s)} + \left(\frac{-s}{1+\exp(-\alpha^2/\lambda - v_2 s)}\right)\left(\frac{1-2\beta}{1+2\beta}\right)^s\right)\\
&= \sum_{s=1}^{n-1} \mathbf{Pr}(S=s)s\left(\frac{1-\exp(\alpha^2/\lambda + v_2 s)\left(\frac{1-2\beta}{1+2\beta}\right)^s}{1+\exp(\alpha^2/\lambda + v_2 s)}\right)\\
&\ge \sum_{s=1}^{n-1} \mathbf{Pr}(S=s)s\left(\frac{1-\exp(\alpha^2/\lambda + v_2 s - 4\beta s)}{1+\exp(\alpha^2/\lambda + v_2 s)}\right).
\end{aligned}$$

6. S is the sum of an odd number of ± 1's, and thus cannot be even.

Assume for contradiction that $v_2 < 1/(n-1)$. Then,

$$\begin{aligned}
&v_2\lambda(n-1)\\
&\geq \sum_{s=1}^{n-1} \mathbf{Pr}(S=s)s\left(\frac{1-\exp(\alpha^2/\lambda + s/(n-1) - 4\beta s)}{1+\exp(\alpha^2/\lambda + s/(n-1))}\right)\\
&\geq \sum_{s=1}^{n-1} \mathbf{Pr}(S=s)s\left(\frac{1-\exp(s/(n-1) - 3\beta s)}{1+\exp(\beta^2\lambda + s/(n-1))}\right) \quad \text{(since } \alpha \leq \beta\lambda\text{)}\\
&\geq \sum_{s=1}^{n-1} \mathbf{Pr}(S=s)s\left(\frac{1-\exp(-2\beta s)}{1+\exp(\beta^2\lambda + s/(n-1))}\right) \quad \text{(for large enough } n\text{)}\\
&\geq \sum_{s\in[\beta(n-1),3\beta(n-1)]} \mathbf{Pr}(S=s)s\left(\frac{1-\exp(-2\beta s)}{1+\exp(\beta^2\lambda + s/(n-1))}\right),
\end{aligned}$$

since each term is positive. Taking the worst-case among $[\beta(n-1), 3\beta(n-1)]$ for each instance of s, and applying Lemma 52, we get

$$\begin{aligned}
v_2 &\geq \frac{1}{\lambda(n-1)} \times \frac{1}{13} \times \beta(n-1)\left(\frac{1-\exp(-2\beta^2(n-1))}{1+\exp(\beta^2\lambda + 3\beta)}\right)\\
&= \frac{30\sqrt{n-1}}{130}\left(\frac{1-\exp(-1/50)}{1+\exp(3/(10\sqrt{n-1}) + 1/(3000n(n-1)))}\right). \qquad (51)
\end{aligned}$$

Thus $v_2 = \Omega(\sqrt{n-1})$, which, for large enough n, contradicts our assumption that $v_2 < 1/(n-1)$, completing the proof. ∎

Not that even with the many approximations made, Inequality (51) gives the desired contradiction at $n = 60$. Even when the weaker bound of $1/50$ discussed following Lemma 52 is used, $n = 145$ still suffices to give the desired contradiction.

Now we're ready to put everything together.

Proof (of Theorem 24): Recall that, by Lemma 46, if $v_1 < 2\beta(n-1)v_2$, then

$$\mathrm{er}_P(\mathbf{v}(P,\lambda)) \geq 3/10.$$

Lemma 51 gives $v_1 < \beta$. Lemma 53 implies $(n-1)v_2 \geq 1$. Therefore $v_1 < \beta(n-1)v_2$, completing the proof. ∎

Using the $1/50$ version of Lemma 52 leads to a proof of the theorem for all even $n \geq 300$.

References

J. Abernethy, C. Lee, A. Sinha, and A. Tewari. Online Linear Optimization Via Smoothing. *COLT*, pages 807–823, 2014.

P. Bachman, O. Alsharif, and D. Precup. Learning with Pseudo-ensembles. *NIPS*, 2014.

P. Baldi and P. Sadowski. The Dropout Learning Algorithm. *Artificial intelligence*, 210: 78–122, 2014.

P. L. Bartlett, M. I. Jordan, and J. D. McAuliffe. Convexity, Classification, and Risk Bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006.

L. Breiman. Some Infinity Theory for Predictor Ensembles. *Annals of Statistics*, 32(1): 1–11, 2004.

N. Cesa-Bianchi, A. Conconi, and C. Gentile. A Second-order Perceptron Algorithm. *COLT*, 2002.

G. E. Dahl. Deep Learning How I Did It: Merck 1st Place Interview, 2012. http://blog.kaggle.com.

G. E. Dahl, T. N. Sainath, and G. E. Hinton. Improving Deep Neural Networks for LVCSR Using Rectified Linear Units and Dropout. *ICASSP*, 2013.

A. DasGupta. *Asymptotic Theory of Statistics and Probability*. Springer, 2008.

L. Deng, J. Li, J. Huang, K. Yao, D. Yu, F. Seide, M. L. Seltzer, G. Zweig, X. He, J. Williams, Y. Gong, and A. Acero. Recent Advances in Deep Learning for Speech Research at Microsoft. *ICASSP*, 2013.

J. Duchi, E. Hazan, and Y. Singer. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *JMLR*, 12:2121–2159, 2011.

R. L. Graham, D. E. Knuth, and O. Patashnik. *Concrete Mathematics*. Addison-Wesley, 1989.

D. P. Helmbold and P. M. Long. On the Necessity of Irrelevant Variables. *JMLR*, 13: 2145–2170, 2012.

G. E. Hinton. Dropout: a Simple and Effective Way to Improve Neural Networks, 2012. videolectures.net.

G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving Neural Networks by Preventing Co-adaptation of Feature Detectors, 2012. Arxiv, arXiv:1207.0580v1.

H. J. Kushner and G. G. Yin. *Stochastic Approximation Algorithms and Applications*. Springer, 1997.

P. M. Long and R. A. Servedio. Random Classification Noise Defeats All Convex Potential Boosters. *Machine Learning*, 78(3):287–304, 2010.

E. Slud. Distribution Inequalities for the Binomial Law. *Annals of Probability*, 5:404–412, 1977.

N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks From Overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

T. Van Erven, W. Kotowski, and M. K. Warmuth. Follow the Leader with Dropout Perturbations. *Annual ACM Workshop on Computational Learning Theory*, pages 949–974, 2014.

S. Wager, S. Wang, and P. Liang. Dropout Training as Adaptive Regularization. *NIPS*, 2013.

S. Wager, W. Fithian, S. Wang, and P. S. Liang. Altitude Training: Strong Bounds for Single-layer Dropout. *NIPS*, 2014.

L. Wan, M. Zeiler, S. Zhang, Y. Le Cun, and R. Fergus. Regularization of Neural Networks using DropConnect. In *ICML*, pages 1058–1066, 2013.

S. Wang and C. Manning. Fast Dropout Training. In *ICML*, pages 118–126, 2013.

T. Zhang. Statistical Behavior and Consistency of Classification Methods Based on Convex Risk Minimization. *Annals of Statistics*, 32(1):56–85, 2004.

Journal of Machine Learning Research 16 (2015) 3455-3467 Submitted 9/14; Revised 2/15; Published 12/15

Agnostic Learning of Disjunctions on Symmetric Distributions

Vitaly Feldman[*] VITALY@POST.HARVARD.EDU
IBM Research - Almaden
San Jose, CA

Pravesh Kothari[†] KOTHARI@CS.UTEXAS.EDU
Department of Computer Science
The University of Texas at Austin
Austin, TX

Editor: Nathan Srebro

Abstract

We consider the problem of approximating and learning disjunctions (or equivalently, conjunctions) on symmetric distributions over $\{0,1\}^n$. Symmetric distributions are distributions whose PDF is invariant under any permutation of the variables. We prove that for every symmetric distribution $\mathcal{D}$, there exists a set of $n^{O(\log(1/\epsilon))}$ functions $\mathbb{S}$, such that for every disjunction c, there is function p, expressible as a linear combination of functions in $\mathbb{S}$, such that p ϵ-approximates c in ℓ_1 distance on $\mathcal{D}$ or $\mathbf{E}_{x\sim\mathcal{D}}[|c(x)-p(x)|] \leq \epsilon$. This implies an agnostic learning algorithm for disjunctions on symmetric distributions that runs in time $n^{O(\log(1/\epsilon))}$. The best known previous bound is $n^{O(1/\epsilon^4)}$ and follows from approximation of the more general class of halfspaces (Wimmer, 2010). We also show that there exists a symmetric distribution $\mathcal{D}$, such that the minimum degree of a polynomial that 1/3-approximates the disjunction of all n variables in ℓ_1 distance on $\mathcal{D}$ is $\Omega(\sqrt{n})$. Therefore the learning result above cannot be achieved via ℓ_1-regression with a polynomial basis used in most other agnostic learning algorithms.

Our technique also gives a simple proof that for any product distribution $\mathcal{D}$ and every disjunction c, there exists a polynomial p of degree $O(\log(1/\epsilon))$ such that p ϵ-approximates c in ℓ_1 distance on $\mathcal{D}$. This was first proved by Blais et al. (2008) via a more involved argument.

Keywords: agnostic learning, symmetric distribution, polynomial approximation, regression, disjunction, conjunction, DNF, decision tree

1. Introduction

The goal of an agnostic learning algorithm for a concept class $\mathcal{C}$ is to produce, for any distribution on examples, a hypothesis h whose error on a random example from the distribution is close to the best possible by a concept from C. This model reflects a common empirical approach to learning, where few or no assumptions are made on the process that generates the examples and a limited space of candidate hypothesis functions is searched in an attempt to find the best approximation to the given data.

∗. Corresponding author.
†. Work done while the author was at IBM Research - Almaden.

Agnostic learning of disjunctions (or, equivalently, conjunctions) is a fundamental question in learning theory and a key step in learning algorithms for other concept classes such as DNF formulas and decision trees. Algorithms for this problem, such as the Set Covering Machine (Marchand and Shawe-Taylor, 2002), are also used in practical applications. There is no known efficient algorithm for the problem, in fact the fastest algorithm that does not make any distributional assumptions runs in $2^{\tilde{O}(\sqrt{n})}$ time (Kalai et al., 2008). Polynomial-time learnability is only known when the examples are very close to being consistent with some disjunction (Awasthi et al., 2010).

While the problem appears to be hard, strong hardness results are known only if the hypothesis is restricted to be a disjunction or a linear threshold function (Ben-David et al., 2003; Bshouty and Burroughs, 2006; Feldman et al., 2009, 2012), or for learning using ℓ_1-regression (Klivans and Sherstov, 2010). Weaker, quasi-polynomial lower bounds are known assuming hardness of learning sparse parities with noise (see Section 5) and, very recently, hardness of refuting random SAT formulas (Daniely and Shalev-Shwartz, 2014). It is also well-known that distribution-independent agnostic learning of disjunctions implies PAC learning of DNF expressions (Kearns et al., 1994). Finally, agnostic learning of disjunctions is known to be closely related to the problem of differentially-private release of answers to conjunctive queries (Gupta et al., 2011).

We consider this problem with an additional assumption that example points are distributed according to a symmetric or a product distribution. Symmetric and product distributions are two incomparable classes of distributions that generalize the well-studied uniform distribution. Theoretical study of learning over symmetric distributions was first done by Wimmer (2010) who gave $n^{O(1/\epsilon^4)}$ time agnostic learning algorithm for the class of halfspaces. Agnostic learning of disjunctions over symmetric distributions on $\{0,1\}^n$ also arises naturally in the well-studied problem of privately releasing answers to all short conjunction queries with low average error (Feldman and Kothari, 2014).

1.1 Our Results

We prove that disjunctions (and conjunctions) are learnable agnostically over any symmetric distribution in time $n^{O(\log(1/\epsilon))}$. This matches the well-known upper bound for the uniform distribution. Our proof is based on ℓ_1-approximation of any disjunction by a linear combination of functions from a fixed set of functions. Such approximation directly gives an agnostic learning algorithm via ℓ_1-regression based approach introduced by Kalai et al. (2008).

A natural and commonly used set of basis functions is the set of all monomials on $\{0,1\}^n$ of some bounded degree. It is easy to see that on product distributions with constant bias, disjunctions longer than some constant multiple of $\log(1/\epsilon)$ are ϵ-close to the constant function 1. Therefore, polynomials of degree $O(\log(1/\epsilon))$ suffice for ℓ_1 (or ℓ_2) approximation on such distributions. This simple argument does not work for general product distributions. However it was shown by Blais et al. (2008) that the same degree (up to a constant factor) still suffices in this case. Their argument is based on the analysis of noise sensitivity under product distributions and implies additional interesting results.

Interestingly, it turns out that low-degree polynomials cannot be used to obtain the same result for all symmetric distributions: we show that there exists a symmetric distribution for which disjunctions are no longer ℓ_1-approximated by low-degree polynomials.

Theorem 1 *There exists a symmetric distribution $\mathcal{D}$ such that for $c = x_1 \vee x_2 \vee \cdots \vee x_n$, any polynomial p that satisfies $\mathbf{E}_{x\sim\mathcal{D}}[|c(x) - p(x)|] \leq 1/3$ is of degree $\Omega(\sqrt{n})$.*

To prove this, we consider the standard linear program to find the coefficients of a degree r polynomial that minimizes pointwise error with the disjunction c. The key idea is to observe that an optimal point for the dual can be used to obtain a distribution on which the ℓ_1 *error* of the best fitting polynomial p for c is same as the value of minimum *pointwise error* of any degree r polynomial with respect to c. When c is a symmetric function, one can further observe that the distribution so obtained is in fact symmetric. Combined with the degree lower bound for uniform approximation by polynomials by Klivans and Sherstov (2010), we obtain the result. The details of the proof appear in Section 3.1.

Our approximation for general symmetric distributions is based on a proof that for the special case of the uniform distribution on S_r (the points from $\{-1,1\}^n$ with Hamming weight r), low-degree polynomials still work, namely, for any disjunction c, there is a polynomial p of degree at most $O(\log(1/\epsilon))$ such that the ℓ_1 error $\mathbf{E}_{x\sim S_r}[|c(x) - p(x)|] \leq \epsilon$.

Theorem 2 *For $r \in \{0,\ldots,n\}$, let S_r denote the set of points in $\{0,1\}^n$ that have exactly r 1's and let $\mathcal{D}_r$ denote the uniform distribution on S_r. For every disjunction c and $\epsilon > 0$, there exists a polynomial p of degree at most $O(\log(1/\epsilon))$ such that $\mathbf{E}_{\mathcal{D}_r}[|c(x) - p(x)|] \leq \epsilon$.*

This result can be easily converted to a basis for approximating disjunctions over arbitrary symmetric distributions. All we need is to partition the domain $\{0,1\}^n$ into layers as $\cup_{0\leq r\leq n} S_r$ and use a (different) polynomial for each layer. Formally, the basis now contains functions of the form $\mathrm{IND}(r)\cdot\chi$, where IND is the indicator function of being in layer of Hamming weight r and χ is a monomial of degree $O(\log(1/\epsilon))$. We note that a related strategy, of constructing a collection of functions, one for each layer of the cube was used by Wimmer (2010) to give an $n^{O(1/\epsilon^4)}$ time agnostic learning algorithm for the class of halfspaces on symmetric distributions. However, his proof technique is based on an involved use of representation theory of the symmetric group and is not related to ours.

Our proof technique also gives a simpler proof for the result of Blais et al. (2008) that implies approximation of disjunction by low-degree polynomials on all product distributions.

Theorem 3 *For any disjunction c and product distribution $\mathcal{D}$ on $\{0,1\}^n$, there is a polynomial p of degree $O(\log(1/\epsilon))$ such that $\mathbf{E}_{x\sim\mathcal{D}}[|c(x) - p(x)|] \leq \epsilon$.*

1.2 Applications

Theorem 2 together with a standard application of ℓ_1 regression (Kalai et al., 2008) yields an agnostic learning algorithm for the class of disjunctions running in time $n^{O(\log(1/\epsilon))}$.

Corollary 4 *There is an algorithm that agnostically learns the class of disjunctions on arbitrary symmetric distributions on $\{0,1\}^n$ in time $n^{O(\log(1/\epsilon))}$.*

This learning algorithm was extended to the class of all coverage functions, and then applied to the well-studied problem of privately releasing answers to all short conjunction queries with low average error (Feldman and Kothari, 2014).

It was shown by Kalai et al. (2009) and Feldman (2010) that agnostic learning of conjunctions over a distribution D in time $T(n, 1/\epsilon)$ implies learning of DNF formulas with s terms over D in time $\mathsf{poly}(n, 1/\epsilon) \cdot T(n, (4s/\epsilon))$. Further, under the same conditions distribution-specific agnostic boosting (Kalai and Kanade, 2009; Feldman, 2010) implies that there exists an agnostic learning algorithm for decision trees with s leaves running in time $\mathsf{poly}(n, 1/\epsilon) \cdot T(n, s/\epsilon)$. Therefore we obtain quasi-polynomial learning algorithms for DNF formulas and decision trees over symmetric distributions.

Corollary 5
1. *DNF formulas with s terms are PAC learnable with error ϵ in time $n^{O(\log(s/\epsilon))}$ over all symmetric distributions;*
2. *Decision trees with s leaves are agnostically learnable with excess error ϵ in time $n^{O(\log(s/\epsilon))}$ over all symmetric distributions.*

We also observe that any algorithm that agnostically learns the class of disjunction on the uniform distribution in time $n^{o(\log(\frac{1}{\epsilon}))}$ would yield a faster algorithm for the notoriously hard problem of Learning Sparse Parities with Noise. This is implicit in prior work (Kalai et al., 2008; Feldman, 2012) and we provide additional details in Section 5.

Dachman-Soled et al. (2015) recently showed that ℓ_1 approximation by polynomials is necessary and sufficient condition for agnostic learning over a product distribution (at least in the statistical query framework of Kearns (1998)). Our agnostic learning algorithm (Theorem 4) and lower bound for polynomial approximation (Theorem 1) demonstrate that this equivalence does not hold for non-product distributions.

2. Preliminaries

We use $\{0,1\}^n$ to denote the n-dimensional Boolean hypercube. Let $[n]$ denote the set $\{1, 2, \ldots, n\}$. For $S \subseteq [n]$, we denote by $\mathsf{OR}_S : \{0,1\}^n \to \{0,1\}$, the monotone Boolean disjunction on variables with indices in S, that is, for any $x \in \{0,1\}^n$, $\mathsf{OR}_S(x) = 0 \Leftrightarrow \forall i \in S \;\; x_i = 0$.

One can define norms and errors with respect to any distribution $\mathcal{D}$ on $\{0,1\}^n$. Thus, for $f : \{0,1\}^n \to \mathbb{R}$, we write the ℓ_1 and ℓ_2 norms of f as $\|f\|_1 = \mathbf{E}_{x\sim\mathcal{D}}[|f(x)|]$ and $\|f\|_2 = \sqrt{\mathbf{E}[f(x)^2]}$ respectively. The ℓ_1 and ℓ_2 error of f with respect to g are given by $\|f - g\|_1$ and $\|f - g\|_2$ respectively.

2.1 Agnostic Learning

The agnostic learning model is formally defined as follows (Haussler, 1992; Kearns et al., 1994).

Definition 6 *Let $\mathcal{F}$ be a class of Boolean functions and let $\mathcal{D}$ be any fixed distribution on $\{0,1\}^n$. For any distribution $\mathcal{P}$ over $\{0,1\}^n \times \{0,1\}$, let opt$(\mathcal{P}, \mathcal{F})$ be defined as: opt$(\mathcal{P}, \mathcal{F}) = \inf_{f\in\mathcal{F}} \mathbf{E}_{(x,y)\sim\mathcal{P}}[|y - f(x)|]$. An algorithm $\mathcal{A}$, is said to agnostically learn $\mathcal{F}$ on $\mathcal{D}$ if for every* excess error $\epsilon > 0$ *and any distribution $\mathcal{P}$ on $\{0,1\}^n \times \{0,1\}$ such that the marginal of $\mathcal{P}$ on*

$\{0,1\}^n$ *is* $\mathcal{D}$*, given access to random independent examples drawn from* $\mathcal{P}$*, with probability at least* $\frac{2}{3}$*,* $\mathcal{A}$ *outputs a hypothesis* $h : \{0,1\}^n \to [0,1]$*, such that* $\mathbf{E}_{(x,y)\sim\mathcal{P}}[|h(x) - y|] \leq opt(\mathcal{P}, \mathcal{F}) + \epsilon$*.*

It is easy to see that given a set of t examples $\{(x^i, y^i)\}_{i\leq t}$ and a set of m functions $\phi_1, \phi_2, \ldots, \phi_m$ finding coefficients $\alpha_1, \ldots, \alpha_m$ which minimize

$$\sum_{i\leq t} \left| \sum_{j\leq m} \alpha_j \phi_j(x^i) - y^i \right|$$

can be formulated as a linear program. This LP is referred to as Least-Absolute-Error (LAE) LP or Least-Absolute-Deviation LP, or ℓ_1 linear regression. As observed by Kalai et al. (2008), ℓ_1 linear regression gives a general technique for agnostic learning of Boolean functions.

Theorem 7 *Let* $\mathcal{C}$ *be a class of Boolean functions,* $\mathcal{D}$ *be distribution on* $\{0,1\}^n$ *and* $\phi_1, \phi_2, \ldots, \phi_m : \{0,1\}^n \to \mathbb{R}$ *be a set of functions that can be evaluated in time polynomial in* n*. Assume that there exists* Δ *such that for each* $f \in \mathcal{C}$*, there exist reals* $\alpha_1, \alpha_2, \ldots, \alpha_m$ *such that*

$$\mathop{\mathbf{E}}_{x\sim\mathcal{D}} \left[\left| \sum_{i\leq m} \alpha_i \phi_i(x) - f(x) \right| \right] \leq \Delta.$$

Then there is an algorithm that for every $\epsilon > 0$ *and any distribution* $\mathcal{P}$ *on* $\{0,1\}^n \times \{0,1\}$ *such that the marginal of* $\mathcal{P}$ *on* $\{0,1\}^n$ *is* $\mathcal{D}$*, given access to random independent examples drawn from* $\mathcal{P}$*, with probability at least* $2/3$*, outputs a function* h *such that*

$$\mathop{\mathbf{E}}_{(x,y)\sim\mathcal{P}}[|h(x) - y|] \leq \Delta + \epsilon.$$

The algorithm uses $O(m/\epsilon^2)$ *examples, runs in time polynomial in* n*,* m*,* $1/\epsilon$ *and returns a linear combination of* ϕ_i*'s.*

The output of this LP is not necessarily a Boolean function but can be converted to a Boolean function with disagreement error of $\Delta+2\epsilon$ using "$h(x) \geq \theta$" function as a hypothesis for an appropriately chosen θ (Kalai et al., 2008).

3. ℓ_1 Approximation on Symmetric Distributions

In this section, we show how to approximate the class of all disjunctions on any symmetric distribution by a linear combination of a small set of basis functions.

As discussed above, polynomials of degree $O(\log(1/\epsilon))$ can ϵ-approximate any disjunction in ℓ_1 distance on any product distribution. This is equivalent to using low-degree monomials as basis functions. We first show that this basis would not suffice for approximating disjunctions on symmetric distributions. Indeed, we construct a symmetric distribution on $\{0,1\}^n$, on which, any polynomial that approximates the monotone disjunction $c = x_1 \vee x_2 \vee \cdots \vee x_n$ within ℓ_1 error of $1/3$ must be of degree $\Omega(\sqrt{n})$.

3.1 Lower Bound on ℓ_1 Approximation by Low-Degree Polynomials

In this section we give the proof of Theorem 1.

Proof [of Thm. 1] Let $d : [n] \to \{0,1\}$ be the predicate corresponding to the disjunction $x_1 \vee x_2 \vee \cdots \vee x_n$, that is, $d(0) = 0$ and $d(i) = 1$ for each $i > 0$.

Consider a natural linear program to find a univariate polynomial f of degree at most d such that $\|d - f\|_\infty = \max_{0 \le i \le n} |d(i) - f(i)|$ is minimized:

$$\begin{aligned} &\min\ \epsilon \\ &s.t.\ \epsilon \ge |d(m) - \sum_{i=0}^{r} \alpha_i \cdot m^i| && \forall\ m \in \{0,\ldots,n\} \\ &\alpha_i \in \mathbb{R} && \forall\ i \in \{0,\ldots,r\}. \end{aligned}$$

This program (and its dual) often comes up in proving polynomial degree lower bounds for various function classes (for example, Sherstov, 2009). If $\{\alpha_0, \alpha_1, \ldots, \alpha_n\}$ is a solution for the program above that has value ϵ then $f(m) = \sum_{i=0}^{r} \alpha_i m^i$ is a degree r polynomial that approximates d within an error of at most ϵ at every point in $\{0,\ldots,n\}$. Klivans and Sherstov (2010) show that there exists an $r^* = \Theta(\sqrt{n})$, such that the optimal value of the program above for $r = r^*$ is $\epsilon^* \ge 1/3$. Standard manipulations can be used to produce the dual of the program:

$$\begin{aligned} &\max\ \sum_{m=0}^{n} \beta_m \cdot d(m) \\ &s.t.\ \sum_{m=0}^{n} \beta_m \cdot m^i = 0 && \forall\ i \in \{0,\ldots,r\} \\ &\sum_{m=0}^{n} |\beta_m| \le 1 \\ &\beta_m \in \mathbb{R} && \forall\ m \in \{0,\ldots,n\}. \end{aligned}$$

Let $\beta^* = \{\beta^*_m\}_{m \in \{0,\ldots,n\}}$ denote an optimal solution for the dual program with $r = r^*$. Then, by strong duality, the value of the dual is also ϵ^*. Observe that $\sum_{m=0}^{n} |\beta^*_m| = 1$, since otherwise we can scale up all the β^*_m by the same factor and increase the value of the program while still satisfying the constraints.

Let $\rho : \{0,\ldots,n\} \to [0,1]$ be defined by $\rho(m) = |\beta^*_m|$. Then ρ can be viewed as a density function of a distribution on $\{0,\ldots,n\}$ and we use it to define a symmetric distribution $\mathcal{D}$ on $\{-1,1\}^n$ as follows: $\mathcal{D}(x) = \rho(w(x))/\binom{n}{w(x)}$, where $w(x) = \sum_{i=1}^{n} x_i$ is the Hamming weight of point x. We now show that any polynomial p of degree r^* satisfies $\mathbf{E}_{x \sim \mathcal{D}}[|c(x) - p(x)|] \ge 1/3$.

We now extract a univariate polynomial f_p that approximates d on the distribution with the density function ρ using p. Let $p_{avg} : \{-1,1\}^n \to \mathbb{R}$ be obtained by averaging p over every layer. That is, $p_{avg}(x) = \mathbf{E}_{z \sim \mathcal{D}_{w(x)}}[p(z)]$, where $w(x)$ denotes the Hamming weight of x. It is easy to check that since c is symmetric, p_{avg} is at least as close to c as p in ℓ_1 distance.

Further, p_{avg} is a symmetric function computed by a multivariate polynomial of degree at most r^* on $\{0,1\}^n$. Thus, the function $f_p(m)$ that gives the value of p_{avg} on points of Hamming weight m can be computed by a univariate polynomial of degree r^*. Further,

$$\mathop{\mathbf{E}}_{x\sim\mathcal{D}}[|c(x)-p(x)|] \geq \mathop{\mathbf{E}}_{x\sim\mathcal{D}}[|c(x)-p_{avg}(x)|] = \mathop{\mathbf{E}}_{m\sim\rho}[|d(m)-f_p(m)|].$$

Let us now estimate the error of f_p w.r.t d on the distribution ρ. Using the fact that f_p is of degree at most r^* and thus $\sum_{m=0}^{n} f_p(m)\cdot\beta_m = 0$ (enforced by the dual constraints), we have:

$$\begin{aligned}\mathop{\mathbf{E}}_{m\sim\rho}[|d(m)-f_p(m)|] &\geq \mathop{\mathbf{E}}_{m\sim\rho}[(d(m)-f_p(m))\cdot \mathsf{sign}(\beta^*_m)]\\ &= \sum_{m=0}^{n} d(m)\cdot\beta^*_m - \sum_{m=0}^{n} f_p(m)\cdot\beta^*_m\\ &= \epsilon^* - 0 = \epsilon^* \geq 1/3.\end{aligned}$$

Thus, the degree of any polynomial that approximates c on the distribution $\mathcal{D}$ with error of at most $1/3$ is $\Omega(\sqrt{n})$. ■

3.2 Upper Bound

In this section, we describe how to approximate disjunctions on any symmetric distribution by using a linear combination of functions from a set of small size. Recall that S_r denotes the set of all points from $\{0,1\}^n$ with weight r.

As we have seen above, symmetric distributions can behave very differently when compared to (constant bounded) product distributions. However, for the special case of the uniform distribution on S_r, denoted by $\mathcal{D}_r$, we show that for every disjunction c, there is a polynomial of degree $O(\log(1/\epsilon))$ that ϵ-approximates it in ℓ_1 distance on $\mathcal{D}_r$. As described in Section 1.1, one can stitch together polynomial approximations on each S_r to build a set of basis functions $\mathbb{S}$ such that every disjunction is well approximated by some linear combination of functions in $\mathbb{S}$. Thus, our goal is now reduced to constructing approximating polynomials on $\mathcal{D}_r$.

Proof [of Thm. 2] We first assume that c is monotone and without loss of generality $c = x_1\vee\cdots\vee x_k$. We will also prove a slightly stronger claim that $\mathbf{E}_{\mathcal{D}_r}[|c(x)-p(x)|] \leq \mathbf{E}_{\mathcal{D}_r}[(c(x)-p(x))^2] \leq \epsilon$ in this case. Let $d : \{0,\ldots,k\}\to\{0,1\}$ be the predicate associated with the disjunction, that is $d(i) = 1$ whenever $i \geq 1$. Note that $c(x) = d\left(\sum_{i\in[k]} x_i\right)$. Therefore our goal is to find a univariate polynomial f that approximates d and then substitute $p_f(x) = f\left(\sum_{i\in[k]} x_i\right)$. This substitution preserves the total degree of the polynomial. We break our construction into several cases based on the relative magnitudes of r, k and ϵ.

If $k \leq 2\ln(1/\epsilon)$, then the univariate polynomial that exactly computes the predicate d satisfies the requirements. Thus assume that $k > 2\ln(1/\epsilon)$. If $r > n-k$, then, c always takes the value 1 on S_r and thus the constant polynomial 1 achieves zero error. If on the

other hand, if $r \geq (n/k)\ln(1/\epsilon)$, then,

$$\mathop{\mathbf{Pr}}_{x\sim\mathcal{D}_r}[c(x)=0] = \frac{\binom{n-k}{r}}{\binom{n}{r}} = \prod_{i=0}^{r-1}\left(1-\frac{k}{n-i}\right) \leq (1-k/n)^r \leq e^{-kr/n} \leq \epsilon.$$

In this case, the constant polynomial 1 achieves an ℓ_2^2 error of at most $\mathbf{Pr}_{x\sim\mathcal{D}_r}[c(x)=0]\cdot 1 \leq \epsilon$. Finally, observe that $r \leq (n/k)\ln(1/\epsilon)$ and $k > 2\ln(1/\epsilon)$ implies $r \leq n/2$. Thus, for the remaining part of the proof, assume that $r < \min\{n-k, (n/k)\ln(1/\epsilon), n/2\}$.

Consider the univariate polynomial $f : \{0,\ldots,k\} \to \mathbb{R}$ of degree t (for some t to be chosen later) that computes the predicate d exactly on $\{0,\ldots,t\}$. This polynomial is given by

$$f(w) = 1 - \frac{1}{t!}\prod_{i=1}^{t}(w-i) = \begin{cases} 1-\binom{w}{t} \text{ for } w>t \\ 1 \text{ for } 0<w\leq t \\ 0 \text{ for } w=0 \end{cases}$$

Let

$$\delta_j = \mathop{\mathbf{Pr}}_{x\sim\mathcal{D}_r}[|\{i \mid x_i=1\}| = j] = \frac{\binom{n-k}{r-j}\cdot\binom{k}{j}}{\binom{n}{r}}.$$

The ℓ_2^2 error of $p_f(x)$ on c satisfies,

$$||p_f - c||_2^2 = \mathop{\mathbf{E}}_{x\sim\mathcal{D}_r}[(c(x)-p_f(x))^2] = \sum_{j=t+1}^{k}\delta_j\cdot\binom{j}{t}^2.$$

We denote the RHS of this equality by $\|d-f\|_2^2$.

We first upper bound δ_j as follows:

$$\begin{aligned}
\delta_j = \frac{\binom{n-k}{r-j}\cdot\binom{k}{j}}{\binom{n}{r}} &= \frac{(n-k)!}{(n-k-r+j)!(r-j)!}\cdot\frac{k!}{(k-j)!j!}\cdot\frac{(n-r)!r!}{n!} \\
&= \frac{1}{j!}\cdot\frac{r!}{(r-j)!}\cdot\frac{k!}{(k-j)!}\cdot\frac{(n-r)!}{n!}\cdot\frac{(n-k)!}{(n-k-r+j)!} \\
&\leq \frac{1}{j!}\cdot(rk)^j\cdot\frac{(n-k)\cdot(n-k-1)\cdots(n-k-r+j+1)}{n\cdot(n-1)\cdots(n-r+1)} \\
&\leq \frac{1}{j!}\cdot(n\ln(1/\epsilon))^j\cdot\frac{1}{(n-r+j)\cdot(n-r+j-1)\cdots(n-r+1)},
\end{aligned}$$

where, in the second to last inequality, we used that $r < n/k\ln(1/\epsilon)$ to conclude that $rk \leq (n\ln(1/\epsilon))$. Now, $r < n/2$ and thus $(n-r+1) > n/2$. Therefore,

$$\delta_j \leq \frac{2^j\cdot(n\ln(1/\epsilon))^j}{n^j\cdot j!} = \frac{(2\ln(1/\epsilon))^j}{j!},$$

and thus:

$$\|d-f\|_2^2 \leq \sum_{j=t+1}^{k}\binom{j}{t}^2\frac{(2\ln(1/\epsilon))^j}{j!}.$$

Set $t = 8e^2\ln(1/\epsilon)$. Using $j! > (j/e)^j > (t/e)^j$ for every $j \geq t+1$, we obtain:

$$\|d-f\|_2^2 \leq \sum_{j=t+1}^{k} 2^{2j} \cdot \left(\frac{2\ln(1/\epsilon)}{8e\ln(1/\epsilon)}\right)^j \leq \epsilon \cdot \sum_{j=t+1}^{\infty} 1/e^j \leq \epsilon. \tag{1}$$

To see that $\mathbf{E}_{\mathcal{D}_r}[|c(x)-p(x)|] \leq \mathbf{E}_{\mathcal{D}_r}[(c(x)-p(x))^2]$ we note that in all cases and for all x, $|p(x)-c(x)|$ is either 0 or ≥ 1. This completes the proof of the monotone case.

We next consider the more general case when $c = x_1 \vee x_2 \vee \cdots \vee x_{k_1} \vee \bar{x}_{k_1+1} \vee \bar{x}_{k_1+2} \vee \cdots \vee \bar{x}_{k_1+k_2}$. Let $c_1 = x_1 \vee x_2 \vee \cdots \vee x_{k_1}$ and $c_2 = \bar{x}_{k_1+1} \vee \bar{x}_{k_1+2} \vee \cdots \vee \bar{x}_{k_1+k_2}$ and $k = k_1 + k_2$. Observe that $c = 1-(1-c_1)\cdot(1-c_2) = c_1 + c_2 - c_1c_2$.

Let p_1 be a polynomial of degree $O(\log(1/\epsilon))$ such that $\|c_1 - p_1\|_1 \leq \|c_1 - p_1\|_2^2 \leq \epsilon/3$. Note that if we swap 0 and 1 in $\{0,1\}^n$ then c_2 will be equal to a monotone disjunction $\bar{c}_2 = x_{k_1+1} \vee x_{k_1+2} \vee \cdots \vee x_{k_1+k_2}$ and $\mathcal{D}_r$ will become $\mathcal{D}_{n-r}$. Therefore by the argument for the monotone case, there exists a polynomial $\bar{p}_2$ of degree $O(\log(1/\epsilon))$ such that $\|\bar{c}_2 - \bar{p}_2\|_1 \leq \epsilon/3$. By renaming the variables back we will obtain a polynomial p_2 of degree $O(\log(1/\epsilon))$ such that $\|c_2 - p_2\|_1 \leq \|c_2 - p_2\|_2^2 \leq \epsilon/3$. Now let $p = p_1 + p_2 - p_1p_2$. Clearly the degree of p is $O(\log(1/\epsilon))$. We now show that $\|c-p\|_1 \leq \epsilon$:

$$\begin{aligned}
\mathop{\mathbf{E}}_{x\sim\mathcal{D}_r}[|c(x)-p(x)|] &= \mathop{\mathbf{E}}_{x\sim\mathcal{D}_r}[|(1-c(x))-(1-p(x))|] \\
&= \mathop{\mathbf{E}}_{x\sim\mathcal{D}_r}[|(1-c_1)(1-c_2)-(1-p_1)(1-p_2)|] \\
&= \mathop{\mathbf{E}}_{x\sim\mathcal{D}_r}[|(1-c_1)(p_2-c_2)+(1-c_2)(p_1-c_1)-(c_1-p_1)(c_2-p_2)|] \\
&\leq \mathop{\mathbf{E}}_{x\sim\mathcal{D}_r}[|(1-c_1)(p_2-c_2)|] + \mathop{\mathbf{E}}_{x\sim\mathcal{D}_r}[|(1-c_2)(p_1-c_1)|] + \mathop{\mathbf{E}}_{x\sim\mathcal{D}_r}[|(c_1-p_1)(c_2-p_2)|] \\
&\leq \mathop{\mathbf{E}}_{x\sim\mathcal{D}_r}[|p_2-c_2|] + \mathop{\mathbf{E}}_{x\sim\mathcal{D}_r}[|p_1-c_1|] + \sqrt{\mathop{\mathbf{E}}_{x\sim\mathcal{D}_r}[(c_1-p_1)^2]\mathop{\mathbf{E}}_{x\sim\mathcal{D}_r}[(c_2-p_2)^2]} \\
&\leq \epsilon/3 + \epsilon/3 + \epsilon/3 = \epsilon.
\end{aligned}$$

■

4. Polynomial Approximation on Product Distributions

In this section, we show that for every product distribution $\mathcal{D} = \prod_{i\in[n]} \mathcal{D}_i$, every $\epsilon > 0$ and every disjunction (or conjunction) c of length k, there exists a polynomial $p : \{0,1\}^n \to \mathbb{R}$ of degree $O(\log(1/\epsilon))$ such that p ϵ-approximates c in ℓ_1 distance on $\mathcal{D}$.

Proof [of Thm. 3] First, we note that without loss of generality we can assume that the disjunction c is equal to $x_1 \vee x_2 \vee \cdots \vee x_k$ for some $k \in [n]$. We can assume monotonicity since we can convert negated variables to un-negated variables by swapping the roles of 0 and 1 for that variable. The obtained distribution will remain product after this operation. Further we can assume that $k = n$ since variables with indices $i > k$ do not affect probabilities of variables with indices $\leq k$ or the value of $c(x)$.

We first note that we can assume that $\mathbf{Pr}_{x\sim\mathcal{D}}[x = 0^k] > \epsilon$ since, otherwise, the constant polynomial 1 gives the desired approximation. Let $\mu_i = \mathbf{Pr}_{x_i\sim\mathcal{D}^i}[x_i = 1]$. Since c is a

symmetric function, its value at any $x \in \{0,1\}^k$ depends only on the Hamming weight of x that we denote by $w(x)$. Thus, we can equivalently work with the univariate predicate $d : \{0,1,\ldots,k\} \to \{0,1\}$, where $d(i) = 1$ for $i > 0$ and $d(0) = 0$.

As in the proof of Theorem 2, we will approximate d by a univariate polynomial f and then use the polynomial $p_f(x) = f(w(x))$ to approximate c.

Let $f : \{0,1,\ldots,k\} \to \mathbb{R}$ be the univariate polynomial of degree t that matches d on all points in $\{0,1,\ldots,t\}$. Thus,

$$f(w) = 1 - \frac{1}{t!} \cdot \prod_{i=1}^{t} (w-i) = \begin{cases} 1-\binom{w}{t} \text{ for } w > t \\ 1 \text{ for } 0<w\leq t \\ 0 \text{ for } w=0 \end{cases}$$

We have,

$$\mathop{\mathbf{E}}_{x\sim\mathcal{D}_r}[(c(x) - p_f(x))^2] = \sum_{j=0}^{k} \mathop{\mathbf{Pr}}_{x\sim\mathcal{D}}[w(x) = j] \cdot |d(j) - f(j)|$$

and we denote the RHS of this equation by $\|d - f\|_1$.

Then:

$$\begin{aligned} \|d - f\|_1 &= \sum_{j=t+1}^{k} \mathop{\mathbf{Pr}}_{\mathcal{D}}[w(x) = j] \cdot |1 - f(j)| \\ &= \sum_{j=t+1}^{k} \mathop{\mathbf{Pr}}_{\mathcal{D}}[w(x) = j] \cdot \binom{j}{t}. \end{aligned} \tag{2}$$

Let us now estimate $\mathbf{Pr}_{\mathcal{D}}[w(x) = j]$.

$$\begin{aligned} \mathop{\mathbf{Pr}}_{\mathcal{D}}[w(x) = j] &= \sum_{S\subseteq[n],\ |S|=j} \prod_{i\in S} \mu_i \cdot \prod_{i\notin S} (1-\mu_i) \\ &\leq \sum_{S\subseteq[n],\ |S|=j} \prod_{i\in S} \mu_i \end{aligned}$$

Observe that in the expansion of $(\sum_{i=1}^{k} \mu_i)^j$, the term $\prod_{i\in S} \mu_i$ occurs exactly $j!$ times. Thus,

$$\sum_{S\subseteq[n],\ |S|=j} \prod_{i\in S} \mu_i \leq \frac{(\sum_{i=1}^{k} \mu_i)^j}{j!}.$$

Set $\mu_{avg} = \frac{1}{k}\sum_{i=1}^{k} \mu_i$. We have:

$$\epsilon \leq \mathop{\mathbf{Pr}}_{x\sim\mathcal{D}}[x = 0^k] = \prod_{i=1}^{k} (1-\mu_i) \leq \left(1 - \frac{1}{k} \cdot \sum_{i=1}^{k} \mu_i\right)^k = (1-\mu_{avg})^k.$$

Thus, $\mu_{avg} = c/k$ for some $c \leq 2\ln(1/\epsilon)$ whenever $k \geq k_0$ where k_0 is some universal constant. In what follows, assume that $k \geq k_0$. (Otherwise, we can use the polynomial of degree equal to k that exactly computes the predicate d on all points).

We are now ready to upper bound the error $\|d - f\|_1$. From Equation (2), we have:

$$\|d - f\|_1 = \sum_{j=t+1}^{k} \mathop{\mathbf{Pr}}_{\mathcal{D}}[w(x) = j] \cdot \binom{j}{t} \leq \sum_{j=t+1}^{k} \frac{(\sum_{i=1}^{k} \mu_i)^j}{j!} \cdot \binom{j}{t}$$
$$\leq \sum_{j=t+1}^{k} \binom{j}{t} \cdot \frac{(2\ln(1/\epsilon))^j}{j!}$$

Setting $t = 4e^2 \ln(1/\epsilon)$ and using the calculation from Equation (1) in the proof of Thm. 2, we obtain that the error $\|d - f\|_1 \leq \epsilon$. ■

5. Agnostic Learning of Disjunctions

Combining Thm. 7 with the results of the previous section (and the discussion in Section 1.1), we obtain an agnostic learning algorithm for the class of all disjunctions on product and symmetric distributions running in time $n^{O(\log(1/\epsilon))}$.

Corollary 8 (Cor. 4, restated) *There is an algorithm that agnostically learns the class of disjunctions on any product or symmetric distribution on $\{0,1\}^n$ with excess error of at most ϵ in time $n^{O(\log(1/\epsilon))}$.*

We now remark that any algorithm that agnostically learns the class of disjunctions (or conjunctions) on n inputs on the uniform distribution on $\{0,1\}^n$ in time $n^{o(\log(\frac{1}{\epsilon}))}$ would yield a faster algorithm for the notoriously hard problem of Learning Sparse Parities with Noise(SLPN). The reduction is based on the technique implicit in the work of Kalai et al. (2008) and Feldman (2012).

For $S \subseteq [n]$, we use χ_S to denote the parity of inputs with indices in S. Let $\mathcal{U}$ denote the uniform distribution on $\{0,1\}^n$. We say that random examples of a Boolean function f have noise of rate η if the label of a random example equals $f(x)$ with probability $1 - \eta$ and $1 - f(x)$ with probability η.

Problem 1 (Learning Sparse Parities with Noise) *For $\eta \in (0, 1/2)$ and $k \leq n$ the problem of learning k-sparse parities with noise η is the problem of finding (with probability at least $2/3$) the set $S \subseteq [n], |S| \leq k$, given access to random examples with noise of rate η of parity function χ_S.*

The fastest known algorithm for learning k-sparse parities with noise η is a recent breakthrough result of Valiant (2012) which runs in time $O(n^{0.8k}\mathsf{poly}(\frac{1}{1-2\eta}))$.

Kalai et al. (2008) and Feldman (2012) prove hardness of agnostic learning of majorities and conjunctions, respectively, based on correlation of concepts in these classes with parities. We state below this general relationship between correlation with parities and reduction to SLPN given by Feldman et al. (2013).

Lemma 9 *Let $\mathcal{C}$ be a class of Boolean functions on $\{0,1\}^n$. Suppose, there exist $\gamma > 0$ and $k \in \mathbb{N}$ such that for every $S \subseteq [n]$, $|S| \leq k$, there exists a function, $f_S \in \mathcal{C}$, such*

that $|\mathbf{E}_{x\sim\mathcal{U}}[f_S(x)\chi_S(x)]| \geq \gamma(k)$. *If there exists an algorithm* $\mathcal{A}$ *that learns the class* $\mathcal{C}$ *agnostically with excess error* ϵ *in time* $T(n, \frac{1}{\epsilon})$ *then, there exists an algorithm* $\mathcal{A}'$ *that learns* k*-sparse parities with noise* $\eta < 1/2$ *in time* $\mathsf{poly}(n, \frac{1}{(1-2\eta)\gamma(k)}) + 2T(n, \frac{2}{(1-2\eta)\gamma(k)})$.

The correlation between a disjunction and a parity is easy to estimate.

Lemma 10 *For any* $S \subseteq [n]$, $|\mathbf{E}_{x\sim\mathcal{U}}[\mathsf{OR}_S(x)\chi_S(x)]| = \frac{1}{2^{|S|-1}}$.

We thus immediately obtain the following corollary.

Theorem 11 *Suppose there exists an algorithm that learns the class of Boolean disjunctions over the uniform distribution agnostically with excess error of* $\epsilon > 0$ *in time* $T(n, \frac{1}{\epsilon})$. *Then there exists an algorithm that learns* k*-sparse parities with noise* $\eta < \frac{1}{2}$ *in time* $\mathsf{poly}(n, \frac{2^{k-1}}{1-2\eta}) + 2T(n, \frac{2^{k-1}}{1-2\eta})$. *In particular, if* $T(n, \frac{1}{\epsilon}) = n^{o(\log(1/\epsilon))}$, *then, there exists an algorithm to solve* k*-SLPN in time* $n^{o(k)}$.

Thus, any algorithm that is asymptotically faster than the one from Cor. 4 yields a faster algorithm for k-SLPN.

References

P. Awasthi, A. Blum, and O. Sheffet. Improved guarantees for agnostic learning of disjunctions. In *Proceedings of COLT*, pages 359–367, 2010.

S. Ben-David, N. Eiron, and P. M. Long. On the difficulty of approximately maximizing agreements. *Journal of Computer and System Sciences*, 66(3):496–514, 2003.

E. Blais, R. O'Donnell, and K. Wimmer. Polynomial regression under arbitrary product distributions. In *Proceedings of COLT*, pages 193–204, 2008.

N. Bshouty and L. Burroughs. Maximizing agreements and coagnostic learning. *Theoretical Computer Science*, 350(1):24–39, 2006.

D. Dachman-Soled, V. Feldman, L.-Y. Tan, A. Wan, and K. Wimmer. Approximate resilience, monotonicity, and the complexity of agnostic learning. In *Proceedings of SODA*, 2015.

A. Daniely and S. Shalev-Shwartz. Complexity theoretic limitations on learning DNF's. *CoRR*, abs/1404.3378, 2014.

V. Feldman. Distribution-specific agnostic boosting. In *Proceedings of Innovations in Computer Science (ICS)*, pages 241–250, 2010.

V. Feldman. A complete characterization of statistical query learning with applications to evolvability. *Journal of Computer System Sciences*, 78(5):1444–1459, 2012.

V. Feldman and P. Kothari. Learning coverage functions and private release of marginals. In *Proceedings of COLT*, 2014.

V. Feldman, P. Gopalan, S. Khot, and A. Ponuswami. On agnostic learning of parities, monomials and halfspaces. *SIAM Journal on Computing*, 39(2):606–645, 2009.

V. Feldman, V. Guruswami, P. Raghavendra, and Y. Wu. Agnostic learning of monomials by halfspaces is hard. *SIAM Journal on Computing*, 41(6):1558–1590, 2012.

V. Feldman, P. Kothari, and J. Vondrák. Representation, approximation and learning of submodular functions using low-rank decision trees. In *Prooceedings of COLT*, pages 30:711–740, 2013.

A. Gupta, M. Hardt, A. Roth, and J. Ullman. Privately releasing conjunctions and the statistical query barrier. In *Proceedings of STOC*, 2011.

D. Haussler. Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Information and Computation*, 100(1):78–150, 1992.

A. Kalai and V. Kanade. Potential-based agnostic boosting. In *Proceedings of NIPS*, pages 880–888, 2009.

A. Kalai, A. Klivans, Y. Mansour, and R. Servedio. Agnostically learning halfspaces. *SIAM Journal on Computing*, 37(6):1777–1805, 2008.

A. Kalai, V. Kanade, and Y. Mansour. Reliable agnostic learning. In *Proceedings of COLT*, 2009.

M. Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM*, 45(6):983–1006, 1998.

M. Kearns, R. Schapire, and L. Sellie. Toward efficient agnostic learning. *Machine Learning*, 17(2-3):115–141, 1994.

A. Klivans and A. Sherstov. Lower bounds for agnostic learning via approximate rank. *Computational Complexity*, 19(4):581–604, 2010.

M. Marchand and J. Shawe-Taylor. The set covering machine. *Journal of Machine Learning Research*, 3:723–746, 2002.

A. Sherstov. Approximate inclusion-exclusion for arbitrary symmetric functions. *Computational Complexity*, 18(2):219–247, 2009.

G. Valiant. Finding correlations in subquadratic time, with applications to learning parities and juntas. In *Proceedings of FOCS*, 2012.

K. Wimmer. Agnostically learning under permutation invariant distributions. In *Proceedings of FOCS*, pages 113–122, 2010.

Journal of Machine Learning Research 16 (2015) 3469-3473 Submitted 12/14; Revised 5/15; Published 12/15

$\mathbb{S}_n$FFT: A Julia Toolkit for Fourier Analysis of Functions over Permutations

Gregory Plumb[†] GPLUMB@WISC.EDU
Deepti Pachauri[†] PACHAURI@CS.WISC.EDU
Risi Kondor[*∓] RISI@CS.UCHICAGO.EDU
Vikas Singh[‡†] VSINGH@BIOSTAT.WISC.EDU
[†] *Department of Computer Sciences*
[‡] *Department of Biostatistics & Med. Info.*
University of Wisconsin-Madison
Madison, WI 53706 USA
[*] *Department of Computer Sciences*
[∓] *Department of Statistics*
University of Chicago
Chicago, IL 60637 USA

Editor: Antti Honkela

Abstract

$\mathbb{S}_n$FFT is an easy to use software library written in the Julia language to facilitate Fourier analysis on the symmetric group (set of permutations) of degree n, denoted $\mathbb{S}_n$ and make it more easily deployable within statistical machine learning algorithms. Our implementation internally creates the irreducible matrix representations of $\mathbb{S}_n$, and efficiently computes fast Fourier transforms (FFTs) and inverse fast Fourier transforms (iFFTs). Advanced users can achieve scalability and promising practical performance by exploiting various other forms of sparsity. Further, the library also supports the partial inverse Fourier transforms which utilizes the smoothness properties of functions by maintaining only the first few Fourier coefficients. Out of the box, $\mathbb{S}_n$FFT currently offers two non-trivial operations for functions defined on $\mathbb{S}_n$, namely *convolution* and *correlation*. While the potential applicability of $\mathbb{S}_n$FFT is fairly broad, as an example, we show how it can be used for clustering ranked data, where each ranking is modeled as a distribution on $\mathbb{S}_n$.

Keywords: permutations, Fourier analysis, fast Fourier transform, Julia

1. Introduction

Over the last few years, there has been a growing interest in the analysis of data given (or expressed) as a probability distribution over permutations. The set of all possible permutations of n elements constitutes a group called the **symmetric group**, denoted $\mathbb{S}_n$. Several recent solutions to ranking problems, hard combinatorial problems, multi-target tracking and feature point matching tasks (in computer vision) have used harmonic analysis on $\mathbb{S}_n$ to derive more efficient algorithms (Huang et al., 2009; Kondor, 2010; Pachauri et al., 2012). While the idea of generalizing the Fourier transform to non-commutative groups is well established in the Mathematics literature, an easy to use and accessible software library will facilitate the adoption of such concepts within machine learning. In this paper, we describe a Julia based open source library which implements the Fourier transform (and associated functionality) for harmonic analysis of functions defined on $\mathbb{S}_n$.

The implementation can use a multi-core cluster (when available) without any need for low-level message passing interface (MPI) programming.

Harmonic analysis on $\mathbb{S}_n$ is defined via the notion of **representations**. A matrix valued function $\rho\colon \mathbb{S}_n \to \mathbb{C}^{d_\rho \times d_\rho}$ is said to be a d_ρ dimensional representation of the symmetric group if $\rho(\sigma_2)\rho(\sigma_1) = \rho(\sigma_2\sigma_1)$ for any pair of permutations $\sigma_1, \sigma_2 \in \mathbb{S}_n$. A representation ρ is said to be *reducible* if there exists a unitary basis transformation which simultaneously block diagonalizes each $\rho(\sigma)$ matrix into a direct sum of lower dimensional representations. If ρ is not reducible, then it is said to be *irreducible*. Irreducible representations or irreps are the elementary building blocks of all of $\mathbb{S}_n$'s representations. A complete set of inequivalent irreducible representations are denoted by $\mathcal{R}$. The Fourier transform of a function $f : \mathbb{S}_n \to \mathbb{C}$ is then defined as the sequence of matrices

$$\hat{f}(\rho) = \sum_{\sigma \in \mathbb{S}_n} f(\sigma)\rho(\sigma) \qquad \rho \in \mathcal{R}. \tag{1}$$

The inverse transform is

$$f(\sigma) = \frac{1}{n!} \sum_{\rho \in \mathcal{R}} d_\rho \operatorname{tr}\big[\hat{f}(\rho)\, \rho(\sigma)^{-1}\big] \qquad \sigma \in \mathbb{S}_n. \tag{2}$$

Much of the practical interest in Fourier transform can be attributed to various interesting properties of irreps, such as conjugacy and unitarity.

1.1 The Irreducible Representation of $\mathbb{S}_n$

There are several ways to construct irreducible representation of $\mathbb{S}_n$ (Sagan, 2001). One such representation is called Young's orthogonal representation (YOR). The YOR matrices are real and unitary and therefore orthogonal. To benefit from the computational advantages of orthogonal matrices, $\mathbb{S}_n$FFT uses YOR internally. In the online documentation, we provide a short review of the technical background relevant for constructing YORs.

2. $\mathbb{S}_n$FFT Toolkit

$\mathbb{S}_n$FFT is implemented in a high-level programming language called Julia (provided under a MIT license). The most important features of the toolkit are accessibility, extensibility, and performance. The toolkit and the required documentation is available at: `https://github.com/GDPlumb/SnFFT.jl/`.

Accessibility. We placed a great deal of emphasis on the ease of use of the toolkit. This will allow a non-specialist (in harmonic analysis) to utilize the functionality of this library within standard machine learning algorithms, when analyzing data on $\mathbb{S}_n$. In particular, the full functionality of $\mathbb{S}_n$FFT is available simply by loading the package "SnFFT" through Julia's built in package manager. The $\mathbb{S}_n$FFT user manual provides many examples demonstrating the syntax for accessing the various features of $\mathbb{S}_n$FFT and gives a high level overview of the key properties of YOR matrices and the Fourier transform. The minimalist design and coding consistency makes $\mathbb{S}_n$FFT easy to use and modify.

Extensibility. Interoperability is a key component of Julia — it allows easy access to various pre-existing high quality and mature libraries written in many other languages with minimal additional overhead. Therefore, various machine learning libraries can be easily incorporated into $\mathbb{S}_n$FFT projects. For example, C and Fortran functions can be called directly from $\mathbb{S}_n$FFt projects without any "glue" code.

$\mathbb{S}_n$FFT allows access to external libraries written in languages such as Python, Java, and R, by easily passing the data to these libraries.

Finally, Julia code can be called directly from C/C++. As a result, $\mathbb{S}_n$FFT can be used seamlessly within existing machine learning tools as needed.

Parallelism. $\mathbb{S}_n$FFT inherits the parallelism offered by the Julia platform. It allows a multi-processing environment to run a code on multiple processes in separate memory domains concurrently. $\mathbb{S}_n$FFT uses empirically derived rules to determine the trade-off between synchronization overhead for multithread computation and single thread sequential computation and proceeds with the best option. In our implementation, $\mathbb{S}_n$FFT functions are designed to use all worker processes that a user makes available to Julia. This setup allows the user to analyze the data on a single process, on multiple processes on a local machine, or via multiple processes spread across a cluster with essentially no change to the user code beyond initially making the processes available.

Sparsity. For various practical applications, we encounter problems where n is greater than 15. Even storing such data is problematic as $n!$ is ~ 1 trillion. Unless one exploits the smoothness/sparsity properties of f, computation will be intractable. But notice that often, problems exhibit interesting sparsity patterns (Kueh et al., 1999); for example, the Fourier transform of functions on homogeneous spaces of $\mathbb{S}_n$ are usually band-limited in the sense that their Fourier transform is identically zero except for a small set of Fourier matrices. $\mathbb{S}_n$FFT is designed to utilize such patterns, making it very efficient. Specifically, the function `sn_fft_bl()` is implemented to offer significant efficiency benefits when the user a priori knows the band-limited form of f. For problems with unknown sparsity pattern, the special function `sn_fft_sp()` first determines the sparsity structure of f and then proceeds to the actual FFT calculation. Partial inverse Fourier transform is also supported in $\mathbb{S}_n$FFT which is important to induce smoothness in f. In particular, function `sn_ifft_p()` can be used to approximate f using just first few Fourier coefficients of the full Fourier transform.

2.1 Related Libraries

An existing library, $\mathbb{S}_n$`ob` described by (Kondor, 2006), motivated the work presented here and offers some of $\mathbb{S}_n$FFT's functionality but support for band-limited behavior of functions is limited in (Kondor, 2006). Further, our Julia implementation gives seamless access to both single and multiple processes and is easier to modify and extend. We believe that such parallelization features will be useful for scalability and integration within machine learning applications.

3. Example: Fourier Domain Features for Clustering Ranks

Consider a ranking dataset composed of N examples where the i^{th} instance ($i = 1, \cdots, N$), is a permutation $\sigma_i \in \mathbb{S}_n$ of n items, listed in order of preference. Given such data, we want to identify groups of examples with similar preferences, which may be helpful for a downstream preference behavior study or rank prediction applications, e.g., (Crammer et al., 2001). Various probabilistic models for ranking are popular in the research community such as Mallows model (Murphy and Martin, 2003), which nicely capture the variability in the observations when the observed rankings are noisy or incomplete (Busse et al., 2007). Typically, the i^{th} instance is represented as a function $f_i(\sigma) = \frac{e^{-\gamma d(\sigma_i,\sigma)}}{Z_\gamma}$ on $\mathbb{S}_n$. Here, γ is the spread parameter, $d(.,.)$ is a valid distance metric on permutations, and Z_γ is the normalization constant. The clustering problem seeks to partition the dataset into K clusters to minimize the following objective:

$$\underset{C_1,\ldots,C_K}{\arg\min} \sum_{k=1}^{K} \sum_{1\leq i,j\leq N:(i,j)\in C_k} \|f_i - f_j\|^2 \ . \tag{3}$$

A geometric view of functions defined on $\mathbb{S}_n$ as embedded in the space $[0,1]^{n!}$ quickly becomes intractable and hard to interpret. On the other hand, the seminal work of (Diaconis, 1988) explains how the Fourier coefficients precisely encode the structural properties of the distributions on $\mathbb{S}_n$. Following ideas described in (Diaconis, 1988), recently, (Clémençon et al., 2011) introduced a Fourier space formulation equivalent to (3)

$$= \frac{1}{n!} \sum_{\rho\in\mathcal{R}} d_\rho \sum_{k=1}^{K} \sum_{1\leq i,j\leq N:(i,j)\in C_k} \|\widehat{f}_i(\rho) - \widehat{f}_j(\rho)\|^2_{HS(d_\rho)} \ . \tag{4}$$

Further, they used a specialized feature selection procedure for clustering the induced spectral features as in (Witten and Tibshirani, 2010) and showed that frequently one only needs a few spectral features to explain the clustering choices. In $\mathbb{S}_n$FFT, only a few lines of code are needed to compute the Fourier transforms, convert them into a data matrix, and pass the data matrix to R's `sparcl` library to perform this clustering. The details of the process can be found in the code of `example_clustering()`.

The foregoing example shows that $\mathbb{S}_n$FFT is fairly flexible and can be used with advanced machine learning libraries for data analysis on $\mathbb{S}_n$. Some example applications which may benefit directly in the short term include multi-object tracking (identity management problem) (Kondor et al., 2007), event based modeling for longitudinal measurements (Huang and Alexander, 2012) and deriving image associations for structure from motion (Pachauri et al., 2014). Some of these applications are described in more detail in the documentation.

Acknowledgments

This work was supported in part by NSF CCF 1320344, NSF CCF 1320755, a REU supplement to NSF RI 1116584 and the University of Wisconsin Graduate School.

References

L. M. Busse, P. Orbanz, and J. M. Buhmann. Cluster analysis of heterogeneous rank data. In *ICML*, 2007.

S. Clémençon, R. Gaudel, and J. Jakubowicz. Clustering rankings in the Fourier domain. In *ECML*, 2011.

K. Crammer, Y. Singer, et al. Pranking with ranking. In *NIPS*, volume 14, 2001.

P. Diaconis. Group representations in probability and statistics. *Institute of Mathematical Statistics Monograph Series*, 1988.

J. Huang and D. Alexander. Probabilistic event cascades for Alzheimer's disease. In *NIPS*, 2012.

J. Huang, C. Guestrin, and L. Guibas. Fourier theoretic probabilistic inference over permutations. *JMLR*, 10, 2009.

R. Kondor. $\mathbb{S}_n$`ob`: a C++ library for fast Fourier transforms on the symmetric group. Downloadable from `http://people.cs.uchicago.edu/~risi/SnOB/index.html`, 2006.

R. Kondor. A Fourier space algorithm for solving quadratic assignment problems. In *SODA*, 2010.

R. Kondor, A. Howard, and T. Jebara. Multi-object tracking with representations of the symmetric group. In *AISTATS*, 2007.

K.-L. Kueh, T. Olson, D. Rockmore, and K.-S. Tan. Nonlinear approximation theory on finite groups. *Department of Mathematics, Dartmouth College, Tech. Rep. PMA-TR99-191*, 1999.

T. B. Murphy and D. Martin. Mixtures of distance-based models for ranking data. *Computational Statistics & Data Analysis*, 41, 2003.

D. Pachauri, M. Collins, V. Singh, and R. Kondor. Incorporating domain knowledge in matching problems via harmonic analysis. In *ICML*, 2012.

D. Pachauri, R. Kondor, and V. Singh. Permutation diffusion maps (PDM) with application to the image association problem in computer vision. In *NIPS*, 2014.

B. E. Sagan. *The Symmetric Group*. Graduate Texts in Mathematics. Springer, 2001.

D. M. Witten and R. Tibshirani. A framework for feature selection in clustering. *Journal of the American Statistical Association*, 105, 2010.

Journal of Machine Learning Research 16 (2015) 3475-3486 Submitted 9/14; Revised 3/15; Published 12/15

The Sample Complexity of Learning Linear Predictors with the Squared Loss

Ohad Shamir OHAD.SHAMIR@WEIZMANN.AC.IL
Department of Computer Science and Applied Mathematics
Weizmann Institute of Science
Rehovot 7610001, Israel

Editor: Mehryar Mohri

Abstract

We provide a tight sample complexity bound for learning bounded-norm linear predictors with respect to the squared loss. Our focus is on an agnostic PAC-style setting, where no assumptions are made on the data distribution beyond boundedness. This contrasts with existing results in the literature, which rely on other distributional assumptions, refer to specific parameter settings, or use other performance measures.

Keywords: sample complexity, squared loss, linear predictors, distribution-free learning

1. Introduction

In machine learning and statistics, the squared loss is the most commonly used loss for measuring real-valued predictions: Given a prediction p and actual target value y, it is defined as $\ell(p, y) = (p - y)^2$. It is intuitive, has a convenient analytical form, and has been extremely well-studied.

In this paper, we concern ourselves with learning bounded-norm linear predictors with respect to the squared loss, in an agnostic PAC learning framework. Formally, for some fixed parameters X, Y, B, we assume the existence of an unknown distribution over $\{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x}\| \leq X\} \times \{y \in \mathbb{R} : |y| \leq Y\}$, from which we are given a training set $S = \{\mathbf{x}_i, y_i\}_{i=1}^m$ of m i.i.d. examples, consisting of pairs of instances $\mathbf{x}$ and target values y. Given a linear predictor $\mathbf{x} \mapsto \langle \mathbf{w}, \mathbf{x} \rangle$, its risk with respect to the squared loss is defined as

$$R(\mathbf{w}) \;=\; \mathbb{E}\left[(\langle \mathbf{w}, \mathbf{x} \rangle - y)^2\right],$$

where the expectation is with respect to $\mathbf{x}, y$. Our goal is to find a linear predictor $\mathbf{w}$ from the hypothesis class of norm-bounded linear predictors,

$$\mathcal{W} = \{\mathbf{w} : \|\mathbf{w}\| \leq B\},$$

such that its excess risk

$$R(\mathbf{w}) - \min_{\mathbf{w} \in \mathcal{W}} R(\mathbf{w})$$

with respect to the best possible predictor in $\mathcal{W}$ is as small as possible. We focus here on the expected excess risk (over the randomness of the training set and algorithm), and study what is the optimal bound on the excess risk one can obtain—also known as a sample complexity

bound—and how it is affected by the problem parameters X, Y, B, d and the sample size m, uniformly over any distribution. Since X and B are invariant to simultaneous scaling (if we re-scale each $\mathbf{x}$ by some factor c, and re-scale each linear predictor $\mathbf{w}$ by $1/c$, all predictions remain the same), we will assume without loss of generality that $X = 1$.

There is a huge literature on learning with the squared loss, with many tight and elegant risk bounds under various assumptions. However, for the framework defined above, there does not appear to be an explicit and self-contained analysis. Much of the existing work (some examples include Hsu et al. (2014); Koltchinskii (2011); Lecué and Mendelson (2014); Tsybakov (2003); Anthony and Bartlett (1999); Lee et al. (1998); Zhang (2005); Audibert and Catoni (2011)) focuses on risk upper bounds, but not lower bounds showing the limits of attainable performance. Moreover, most existing work considers settings different than ours in one or more of the following aspects:

- *Additional Distributional Assumptions:* In our agnostic setting, we make no assumptions on the data distribution except boundedness. In contrast, most existing work relies on additional assumptions. Perhaps the most common assumption is a well-specified model, under which there exists a fixed $\mathbf{w} \in \mathbb{R}^d$ such that $y = \langle \mathbf{w}, \mathbf{x} \rangle + \xi$, where ξ is a zero-mean noise term (such as Tsybakov (2003)). Other works impose additional conditions on the distribution of $\mathbf{x}$ (for example, that the covariance matrix of $\mathbf{x}$ is well behaved, such as Hsu et al. (2014)), or consider a fixed design setting where the data instances $\mathbf{x}$ are not sampled i.i.d.. These assumptions usually lead to excess risk bounds which scale (at least in finite dimensions) as dY^2/m, independent of the norm bound B. However, as we will see later, this is not the behavior in our setting.

- *Bounds not on the excess risk:* Many of the existing results are not on the excess risk, but rather on $\mathbb{E}[\|\mathbf{w} - \mathbf{w}^*\|^2]$ or $\mathbb{E}[(\langle \mathbf{w}, \mathbf{x} \rangle - \langle \mathbf{w}^*, \mathbf{x} \rangle)^2]$, where $\mathbf{w}^* = \arg\min_{\mathbf{w} \in \mathcal{W}} R(\mathbf{w})$ (such as Koltchinskii (2011); Lecué and Mendelson (2014)). The former measure is relevant for parameter estimation, while the latter measure can be shown to equal the excess risk when $\mathbf{w}^* = \arg\min_{\mathbf{w} \in \mathbb{R}^d} R(\mathbf{w})$ (in other words, $B = \infty$, see Lemma 2 below). However, when we deal with the hypothesis class of norm-bounded predictors, then the excess risk can be larger by an arbitrary factor[1]. Therefore, upper bounds on these measures do not imply upper bounds on the excess risk in our setting. We remark that in our distribution-free setting, we must constrain the hypothesis class, since if our hypothesis class contains all linear predictors ($B = \infty$), then the lower bounds below imply that non-trivial learning is impossible with any sample size (regardless of the dimension d).

- *Bounded Functions:* Many learning theory results for the squared loss (such as those based on fat-shattering techniques, see Lee et al. (1998); Anthony and Bartlett (1999)) assume that the predictor functions and target values are bounded in some fixed

1. For example, consider a distribution on (x, y) such that $(x, y) = (1, 1)$ with probability 1, and $\mathcal{W} = \{w : w \in [-1/2, 1/2]\}$. Then clearly, $w^* = 1/2$, and $\mathbb{E}[(wx - w^*x)^2] = \mathbb{E}[(w - w^*)^2] = (1/2 - w)^2$. However, the excess risk equals $(w-1)^2 - (1/2-1)^2 = w^2 - 2w + 3/4 = (1/2-w)^2 + (1/2-w)$. This is larger than the excess risk by an additive factor of $(1/2 - w)$, and a multiplicative factor of $\frac{1}{1/2-w}$—arbitrarily large if w is close to $w^* = 1/2$.

interval (such as [-1,+1]). In our setting, this would correspond to assuming $B, Y \leq 1$. Other results (such as Bartlett and Mendelson (2003)) assume Lipschitz loss functions, which is not satisfied for the squared loss. One notable exception is Srebro et al. (2010), which analyzes smooth and strongly-convex losses (such as the squared loss) and provide tight sample complexity bounds. However, their results apply either when the functions are bounded by 1, or when d is extremely large or infinite dimensional. In contrast, we provide more general results which hold for any d and when the functions are not necessarily bounded by 1.

- *Collapsing Problem Parameters Together:* Some works, such as Srebro et al. (2010), implicitly take Y to equal the largest possible prediction, $\sup_{\mathbf{w},\mathbf{x}} |\langle \mathbf{w}, \mathbf{x} \rangle| = B$, and give results only in terms of B. However, we will see that B and Y affect the excess risk in a different manner, and it is thus important to discern between them. Moreover, B and Y can often have very different magnitudes. For example, in learning problems where the instances $\mathbf{x}$ tend to be sparse, we may want to have the norm bound B of the predictor to scale with the dimension d, while the bound on the target values Y remain a fixed constant.

2. Main Result

Our main result is the following lower bound on the attainable excess risk:

Theorem 1 *There exists a universal constant c, such that for any dimension d, sample size m, target value bound Y, predictor norm bound $B \geq 2Y$, and for any algorithm returning a linear predictor $\hat{\mathbf{w}}$, there exists a data distribution such that*

$$\mathbb{E}[R(\hat{\mathbf{w}}) - R(\mathbf{w}^*)] \geq c \ \min\left\{ Y^2, \frac{B^2 + dY^2}{m}, \frac{BY}{\sqrt{m}} \right\},$$

where $\mathbf{w}^* = \arg\min_{\mathbf{w}:\|\mathbf{w}\| \leq B} R(\mathbf{w})$, and the expectation is with respect to the training set and the (possible) randomness of the algorithm.

Based on existing results in the literature, this bound has essentially matching upper bounds, up to logarithmic factors:

- Using the trivial zero predictor $\hat{\mathbf{w}} = \mathbf{0}$, we are guaranteed that $R(\hat{\mathbf{w}}) - R(\mathbf{w}^*) \leq R(\hat{\mathbf{w}}) = \mathbb{E}[(\langle \mathbf{0}, \mathbf{x} \rangle - y)^2] = \mathbb{E}[y^2] \leq Y^2$.
- Using the Vovk-Azoury-Warmuth forecaster (Vovk (2001); Azoury and Warmuth (2001)) and a standard online-to-batch conversion technique (see for instance Shalev-Shwartz (2012), corollary 5.2), we have an algorithm for which

$$\mathbb{E}[R(\hat{\mathbf{w}}) - R(\mathbf{w}^*)] \leq \mathcal{O}\left(\frac{B^2 + dY^2 \log(1 + m/d)}{m} \right).$$

- Alternatively, by corollary 3 in Srebro et al. (2010) [2], using mirror descent with an online-to-batch conversion gives us an algorithm for which $\mathbb{E}[R(\hat{\mathbf{w}}) - R(\mathbf{w}^*)] \leq$

2. Where $\bar{L}^* \leq Y^2$ and $H = 2$ for the squared loss.

$\mathcal{O}\left(\frac{BY}{\sqrt{m}} + \frac{B^2}{m}\right)$. In the regime where this bound is smaller than Y^2, it can be verified that $BY/\sqrt{m}$ is the dominant term, in which case we get an $\mathcal{O}(BY/\sqrt{m})$ bound.

Taking the best of these algorithmic approaches, we get the minimum of these upper bounds, i.e. we can find a predictor $\hat{\mathbf{w}}$ for which

$$\mathbb{E}[R(\hat{\mathbf{w}}) - R(\mathbf{w}^*)] \leq \mathcal{O}\left(\min\left\{Y^2, \frac{B^2 + dY^2 \log\left(1 + \frac{m}{d}\right)}{m}, \frac{BY}{\sqrt{m}}\right\}\right).$$

We conjecture that the same bound, perhaps up to log-factors, can be shown for empirical risk minimization (i.e. given a training set $\{(\mathbf{x}_i, y_i)\}_{i=1}^m$, return $\hat{\mathbf{w}} = \min_{\mathbf{w}:\|\mathbf{w}\|\leq B} \frac{1}{m}\sum_{i=1}^m (\langle \mathbf{w}, \mathbf{x}_i\rangle - y_i)^2$).

This result has some interesting consequences: First, it implies that even when $d = 1$ (i.e. a one-dimensional problem), there is a non-trivial dependence on the norm bound B. This is in contrast to results under the well-specified model or other common distributional assumptions, which lead to upper bounds independent of B. Second, it shows that in a finite-dimensional setting, although the squared loss $(\langle \mathbf{w}, \mathbf{x}\rangle - y)^2$ may appear symmetric with respect to y and $\langle \mathbf{w}, \mathbf{x}\rangle$, the attainable excess risk is actually much more sensitive to the bound Y on $|y|$ than to the bound B on $|\langle \mathbf{w}, \mathbf{x}\rangle|$, due to the d factor. For example, if Y is a constant, then B can be as large as the dimension d without affecting the leading term of the excess risk. Third, in the context of online learning, it implies that the Vovk-Azoury-Warmuth forecaster is essentially optimal in our setting and for a finite-dimensional regime, in terms of its dependence on both d and B (the lower bounds in Vovk (2001); Singer et al. (2002) do not show an explicit dependence on B).

3. Proof of Thm. 1

The proof of our main result consist of two separate lower bounds, each of which uses a different construction. The theorem follows by combining them and performing a few simplifications.

We begin by recalling the following result, which follows from the well-known orthogonality principle:

Lemma 2 *Let $R(\mathbf{w}) = \mathbb{E}[(\langle \mathbf{w}, \mathbf{x}\rangle - y)^2]$, where the expectation is over $\mathbf{x}, y$, and let $\mathbf{w}^* = \arg\min_{\mathbf{w}:\|\mathbf{w}\|\leq B} R(\mathbf{w})$. Then for any $\mathbf{w} : \|\mathbf{w}\| \leq B$, it holds that*

$$R(\mathbf{w}) - R(\mathbf{w}^*) \geq \mathbb{E}[(\langle \mathbf{w}, \mathbf{x}\rangle - \langle \mathbf{w}^*, \mathbf{x}\rangle)^2],$$

with equality when $B = \infty$.

Proof For any $\mathbf{w} \in \mathbb{R}^d$, define the linear function $f_{\mathbf{w}} : \mathbb{R}^d \to \mathbb{R}$ by $f_{\mathbf{w}}(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x}\rangle$. Then $\{f_{\mathbf{w}}(\cdot) : \|\mathbf{w}\| \leq B\}$ corresponds to a closed convex set in the L^2 function space defined via the inner product $\langle f, g\rangle = \mathbb{E}_{\mathbf{x}}[f(\mathbf{x})g(\mathbf{x})]$ and norm $\|f\|^2 = \mathbb{E}_{\mathbf{x}}[f^2(\mathbf{x})]$. Moreover, letting $\eta(\mathbf{x}) = \mathbb{E}[y|\mathbf{x}]$, we have

$$\begin{aligned} R(\mathbf{w}) - R(\mathbf{w}^*) &= \mathbb{E}[(\langle \mathbf{w}, \mathbf{x}\rangle - y)^2] - \mathbb{E}[(\langle \mathbf{w}, \mathbf{x}\rangle - y)^2] \\ &= \mathbb{E}[(f_{\mathbf{w}}(\mathbf{x}) - \eta(\mathbf{x}))^2] - \mathbb{E}[(f_{\mathbf{w}^*}(\mathbf{x}) - \eta(\mathbf{x}))^2] \\ &= \|f_{\mathbf{w}} - \eta\|^2 - \|f_{\mathbf{w}^*} - \eta\|^2. \end{aligned}$$

Moreover,

$$\mathbb{E}[(\langle \mathbf{w}, \mathbf{x}\rangle - \langle \mathbf{w}^*, \mathbf{x}\rangle)^2] = \mathbb{E}[\langle \mathbf{w} - \mathbf{w}^*, \mathbf{x}\rangle^2] = \|f_{\mathbf{w}-\mathbf{w}^*}\|^2 = \|f_{\mathbf{w}} - f_{\mathbf{w}^*}\|^2.$$

Therefore, the inequality in the lemma can be written as

$$\|f_{\mathbf{w}} - \eta\|^2 - \|f_{\mathbf{w}^*} - \eta\|^2 \;\geq\; \|f_{\mathbf{w}} - f_{\mathbf{w}^*}\|^2,$$

or equivalently

$$\|f_{\mathbf{w}} - f_{\mathbf{w}^*}\|^2 + \|f_{\mathbf{w}^*} - \eta\|^2 \;\leq\; \|f_{\mathbf{w}} - \eta\|^2. \tag{1}$$

To see why this inequality hold, recall that the set of linear functionals (which includes $f_{\mathbf{w}}$ and $f_{\mathbf{w}^*}$) form a linear subspace in L^2. Moreover, $f_{\mathbf{w}^*}$ is the projection of η on the set $\{f_{\mathbf{w}} : \|\mathbf{w}\| \leq B\}$: To see this, note that

$$\begin{aligned}
\mathbf{w}^* &= \arg\min_{\mathbf{w}:\|\mathbf{w}\|\leq B} \mathbb{E}[(\langle \mathbf{w}, \mathbf{x}\rangle - y)^2 \\
&= \arg\min_{\mathbf{w}:\|\mathbf{w}\|\leq B} \mathbb{E}\left[\mathbb{E}\left[(\langle \mathbf{w}, \mathbf{x}\rangle - y)^2 | \mathbf{x}\right]\right] \\
&= \arg\min_{\mathbf{w}:\|\mathbf{w}\|\leq B} \mathbb{E}\left[\langle \mathbf{w}, \mathbf{x}\rangle^2 - 2\mathbb{E}\left[\langle \mathbf{w}, y\mathbf{x}\rangle | \mathbf{x}\right] + \mathbb{E}[y^2|\mathbf{x}] \mid \mathbf{x}\right] \\
&= \arg\min_{\mathbf{w}:\|\mathbf{w}\|\leq B} \mathbb{E}\left[\langle \mathbf{w}, \mathbf{x}\rangle^2 - 2\langle \mathbf{w}, \mathbb{E}[y|\mathbf{x}]\mathbf{x}\rangle + \mathbb{E}_y[y|\mathbf{x}]^2 \mid \mathbf{x}\right] \\
&= \arg\min_{\mathbf{w}:\|\mathbf{w}\|\leq B} \mathbb{E}\left[\left(\langle \mathbf{w}, \mathbf{x}\rangle - \eta(\mathbf{x})\right)^2\right],
\end{aligned}$$

(where in the fourth equality we used the fact that adding and subtracting terms independent of $\mathbf{w}$ does not change the argmin), and therefore $f_{\mathbf{w}^*} = \arg\min_{f_{\mathbf{w}}:\|\mathbf{w}\|\leq B} \|f_{\mathbf{w}} - \eta\|^2$. When $B = \infty$, then $f_{\mathbf{w}^*}$ is simply the projection of η on the linear sub-space of linear functionals, hence (1) holds with equality by the Pythagorean theorem (see figure 1). When $B < \infty$, then $f_{\mathbf{w}^*}$ is the projection of η on a constrained convex subset of this linear space, and we only have an inequality. ■

Our first construction provides an excess risk lower bound even when we deal with one-dimensional problems:

Theorem 3 *There exists a universal constant c, such that for any sample size m, target value bound Y, predictor norm bound $B \geq 2Y$, and any algorithm returning a linear predictor $\hat{\mathbf{w}}$, there exists a data distribution in $d = 1$ dimensions such that*

$$\mathbb{E}[R(\hat{w}) - R(w^*)] \geq c \min\left\{Y^2, \frac{B^2}{m}\right\}.$$

The expectation is with respect to the training set and the (possible) randomness of the algorithm.

Proof Let α, γ be small positive parameters in $(0, 1]$ to be chosen later, such that $\alpha > \gamma$, and consider the following two distributions over (x, y):

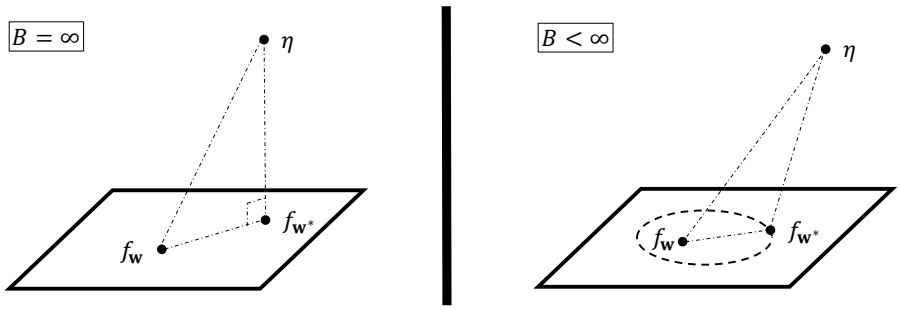


Figure 1: Illustration of inequality in the proof of Lemma 2. The rectangle represents the subspace of linear functionals, and the dotted circle in the right figure represents the convex subset $\{f_{\mathbf{w}} : \|\mathbf{w}\| \leq B\}$.

- Distribution $\mathcal{D}_0$: $y = Y$ w.p. 1; $x = \begin{cases} Y/B & \text{w.p. } \alpha \\ 0 & \text{w.p. } 1-\alpha \end{cases}$.
- Distribution $\mathcal{D}_1$: $y = Y$ w.p. 1; $x = \begin{cases} 1 & \text{w.p. } \gamma \\ Y/B & \text{w.p. } \alpha - \gamma \\ 0 & \text{w.p. } 1-\alpha \end{cases}$.

Note that since $B \geq 2Y$, $|x| \leq 1$, so these are indeed valid distributions. Intuitively, in both distributions x is small most of the time, but under $\mathcal{D}_1$ it can occasionally have a "large" value of 1. Unless the sample size is large enough, it is not possible to distinguish between these two distributions, and this will lead to an excess risk lower bound.

Let $\mathbb{E}_0$ and $\mathbb{E}_1$ denote expectations with respect to $\mathcal{D}_0$ and $\mathcal{D}_1$ respectively. Let

$$w_0^* = B$$

denote the optimal predictor under $\mathcal{D}_0$, and let

$$w_1^* = \frac{\mathbb{E}_1[yx]}{\mathbb{E}_1[x^2]} = \frac{(Y^2/B)(\alpha-\gamma)+Y\gamma}{(Y^2/B^2)(\alpha-\gamma)+\gamma} = B\frac{Y^2(\alpha-\gamma)+BY\gamma}{Y^2(\alpha-\gamma)+B^2\gamma}$$

denote the optimal predictor under $\mathcal{D}_1$. Note that since $B \geq 2Y$, we have $w_1^* \leq w_0^*$, and moreover,

$$\begin{aligned}(w_1^* - w_0^*)^2 = B^2\left(\frac{Y^2(\alpha-\gamma)+BY\gamma}{Y^2(\alpha-\gamma)+B^2\gamma} - 1\right)^2 &= B^4\gamma^2\left(\frac{Y-B}{Y^2\alpha+(B^2-Y^2)\gamma}\right)^2 \\ &\geq B^4\gamma^2\left(\frac{Y-B}{Y^2\alpha+B^2\gamma}\right)^2. \qquad (2)\end{aligned}$$

By Yao's minimax principle, it is sufficient to show that when choosing either $\mathcal{D}_0$ or $\mathcal{D}_1$ uniformly at random, and generating a dataset according to that distribution, any

deterministic algorithm attains the lower bound in the theorem. Using Lemma 2, and the notation $\Pr_0$ (respectively $\Pr_1$) to denote probabilities with respect to $\mathcal{D}_0$ (respectively $\mathcal{D}_1$), we have

$$\begin{aligned}
\mathbb{E}[R(\hat{w})-R(w^*)] &= \frac{1}{2}\left(\mathbb{E}_0[(\hat{w}x - w_0^*x)^2] + \mathbb{E}_1[(\hat{w}x - w_1^*x)^2]\right) \\
&\geq \frac{1}{2}\frac{Y^2\alpha}{B^2}\left(\mathbb{E}_0[(\hat{w} - w_0^*)^2] + \mathbb{E}_1[(\hat{w} - w_1^*)^2]\right) \\
&\geq \frac{1}{2}\frac{Y^2\alpha}{B^2}\left(\frac{w_1^* - w_0^*}{2}\right)^2\left(\Pr_0\left(\hat{w} < \frac{w_0^*+w_1^*}{2}\right) + \Pr_1\left(\hat{w} \geq \frac{w_0^*+w_1^*}{2}\right)\right) \\
&= \frac{1}{2}\frac{Y^2\alpha}{B^2}\left(\frac{w_1^* - w_0^*}{2}\right)^2\left(1 - \left(\Pr_0\left(\hat{w} \geq \frac{w_0^*+w_1^*}{2}\right) - \Pr_1\left(\hat{w} \geq \frac{w_0^*+w_1^*}{2}\right)\right)\right) \\
&\geq \frac{1}{2}\frac{Y^2\alpha}{B^2}\left(\frac{w_1^* - w_0^*}{2}\right)^2\left(1 - \left|\Pr_0\left(\hat{w} \geq \frac{w_0^*+w_1^*}{2}\right) - \Pr_1\left(\hat{w} \geq \frac{w_0^*+w_1^*}{2}\right)\right|\right),
\end{aligned}$$

where in the second inequality we used the fact that $w_1^* \leq w_0^*$. By Pinsker's inequality, since $\hat{w}$ is a deterministic function of the training set S, this is at least

$$\frac{1}{8}\frac{Y^2\alpha}{B^2}\left(w_1^* - w_0^*\right)^2\left(1 - \sqrt{\frac{1}{2}D_{kl}(p_0(S)||p_1(S))}\right),$$

where D_{kl} is the Kullback-Leibler divergence, and p_0 (respectively p_1) is the probability measure of the sample with respect to $\mathcal{D}_0$ (respectively $\mathcal{D}_1$). Since S is composed of m i.i.d. instances, and the target value y is fixed under both distributions, we can invoke the chain rule and rewrite this as

$$\frac{1}{8}\frac{Y^2\alpha}{B^2}\left(w_1^* - w_0^*\right)^2\left(1 - \sqrt{\frac{m}{2}\ D_{kl}(p_0(x)||p_1(x))}\right).$$

To simplify the bound, we use the following fact (see for instance Gibbs and Su (2002), Theorem 5):

Lemma 4 *For any probability distributions p, q over the same discrete sample space, it holds that $D_{kl}(p||q)$ is upper bounded by the χ^2 divergence between p and q, which equals $\sum_a \frac{(p(a)-q(a))^2}{q(a)}$.*

Using this lemma, we have

$$D_{kl}(p_0(x)||p_1(x)) \leq \frac{\gamma^2}{\gamma} + \frac{\gamma^2}{\alpha-\gamma} = \gamma\left(1 + \frac{\gamma}{\alpha-\gamma}\right).$$

Plugging this back, as well as the value of $(w_1^* - w_0^*)^2$ from (2), we get an excess loss lower bound on the form

$$\frac{1}{8}Y^2\alpha B^2\gamma^2\left(\frac{Y-B}{Y^2\alpha + B^2\gamma}\right)^2\left(1 - \sqrt{\frac{m}{2}\gamma\left(1 + \frac{\gamma}{\alpha-\gamma}\right)}\right),$$

We now consider two cases:

- If $m \leq B^2/Y^2$, we pick $\alpha = 1$ and $\gamma = 1/3m$, and get that the expression above is at least

$$\begin{aligned}
&\frac{Y^2}{72}\frac{B^2}{m^2}\left(\frac{B-Y}{Y^2+B^2/3m}\right)^2\left(1-\sqrt{\frac{1}{6}\left(1+\frac{1/3m}{1-1/3m}\right)}\right) \\
&= \frac{Y^2}{72}\left(\frac{B(B-Y)}{mY^2+B^2/3}\right)^2\left(1-\sqrt{\frac{1}{6}\left(1+\frac{1}{3m-1}\right)}\right) \\
&\geq \frac{Y^2}{72}\left(\frac{B(B-Y)}{(B^2/Y^2)Y^2+B^2/3}\right)^2\left(1-\sqrt{\frac{1}{6}\left(1+\frac{1}{3m-1}\right)}\right) \\
&\geq \frac{Y^2}{72}\left(\frac{B(B-Y)}{(1+1/3)B^2}\right)^2\left(1-\sqrt{\frac{1}{6}\left(1+\frac{1}{2}\right)}\right) \\
&\geq 0.003\ Y^2\left(\frac{B-Y}{B}\right)^2 \ = \ 0.003\ Y^2\left(1-\frac{Y}{B}\right)^2 \ \geq \ 0.003\ Y^2\left(1-\frac{1}{2}\right)^2,
\end{aligned}$$

  where we used the assumption that $B \geq 2Y$.

- If $m > B^2/Y^2$, we pick $\alpha = B^2/(Y^2 m)$ and $\gamma = 1/3m$ and get that the expression above is at least

$$\begin{aligned}
&\frac{1}{8}\frac{B^4}{m}\frac{1}{9m^2}\left(\frac{B-Y}{B^2/m+B^2/3m}\right)^2\left(1-\sqrt{\frac{1}{6}\left(1+\frac{1/3m}{(B^2/Y^2-1/3)/m}\right)}\right) \\
&\geq \frac{1}{72}\frac{(B-Y)^2}{m(1+1/3)^2}\left(1-\sqrt{\frac{1}{6}\left(1+\frac{1/3}{4-1/3}\right)}\right) \\
&\geq 0.004\frac{(B-Y)^2}{m} \ \geq \ 0.004\frac{(B-B/2)^2}{m} \ = \ 0.001\frac{B^2}{m},
\end{aligned}$$

  where we used the assumption that $B \geq 2Y$.

Combining the two cases, we get an excess risk lower bound of $c\ \min\left\{Y^2, \frac{B^2}{m}\right\}$ for some universal constant c. ■

Our second construction provides a different type of bound, which quantifies a dependence on the dimension d:

Theorem 5 *There exists a universal constant c, such that for any dimension d, sample size m, target value bound Y, predictor norm bound B and any algorithm returning a linear predictor $\hat{\mathbf{w}}$, there exists a data distribution in d dimensions such that*

$$\mathbb{E}[R(\hat{\mathbf{w}}) - R(\mathbf{w}^*)] \geq c\ \min\left\{Y^2, B^2, \frac{dY^2}{m}, \frac{BY}{\sqrt{m}}\right\}.$$

The expectation is with respect to the training set and the (possible) randomness of the algorithm.

Proof By Yao's minimax principle, it is sufficient to display a randomized choice of data distributions, with respect to which the expected excess error of any deterministic algorithm attains the lower bound in the theorem. In the sequel, we use $\mathbb{E}$ to denote expectation with respect to the random choice of data distribution, as well as the random drawing of a training set from the distribution.

In particular, fix some $d' \leq d$ to be chosen later, let $\boldsymbol{\sigma} \in \{-1,+1\}^{d'}$ be chosen uniformly at random, and consider the distribution $\mathcal{D}_{\boldsymbol{\sigma}}$ (indexed by $\boldsymbol{\sigma}$) over examples $(\mathbf{x}, y)$, defined as follows: $\mathbf{x}$ is chosen uniformly at random among the first d' standard basis vectors. Conditioned on $\mathbf{x} = \mathbf{e}_i$, y is chosen to equal Y with probability $\frac{1}{2}(1+\sigma_i b)$, where $b = \min\{1/2, \sqrt{d'/6m}\}$, and $-Y$ otherwise.

A simple calculation shows that the optimum $\mathbf{w}^* = \arg\min_{\mathbf{w}:\|\mathbf{w}\|\leq B} R(\mathbf{w})$ is such that

$$\forall\, i \in \{1,\ldots,d'\}, \quad w_i^* = \sigma_i \min\{Yb, B/\sqrt{d'}\}.$$

Therefore, using Lemma 2 and the notation $\mathbf{1}_A$ as the indicator function for the event A:

$$\begin{aligned}
\mathbb{E}\left[R(\hat{\mathbf{w}}) - R(\mathbf{w}^*)\right] &\geq \mathbb{E}\left[\mathbb{E}_{\mathbf{x}}[(\langle\hat{\mathbf{w}},\mathbf{x}\rangle - \langle\mathbf{w}^*,\mathbf{x}\rangle)^2]\right] \\
&= \frac{1}{d'}\sum_{i=1}^{d'} \mathbb{E}[(\hat{\mathbf{w}}_i - \mathbf{w}_i^*)^2] \\
&\geq \frac{1}{d'}\sum_{i=1}^{d'} \mathbb{E}[(\mathbf{w}_i^*)^2 \mathbf{1}_{\hat{\mathbf{w}}_i \mathbf{w}_i^* \leq 0}] \\
&= \frac{1}{d'}\left(\min\{Yb, B/\sqrt{d'}\}\right)^2 \sum_{i=1}^{d'} \Pr(\hat{\mathbf{w}}_i \mathbf{w}_i^* \leq 0).
\end{aligned}$$

Since σ_i is uniformly distributed on $\{-1,+1\}$, and has the same sign as w_i^*, this equals

$$\begin{aligned}
&\frac{1}{d'}\left(\min\{Yb, B/\sqrt{d'}\}\right)^2 \sum_{i=1}^{d'} \frac{1}{2}\left(\Pr(\hat{\mathbf{w}}_i \geq 0 | \sigma_i < 0) + \Pr(\hat{\mathbf{w}}_i \leq 0 | \sigma_i > 0)\right) \\
&\geq \frac{1}{2d'}\left(\min\{Yb, B/\sqrt{d'}\}\right)^2 \sum_{i=1}^{d'} \left(1 - \Pr(\hat{\mathbf{w}}_i \leq 0 | \sigma_i < 0) + \Pr(\hat{\mathbf{w}}_i \leq 0 | \sigma_i > 0)\right) \\
&\geq \frac{1}{2d'}\left(\min\{Yb, B/\sqrt{d'}\}\right)^2 \sum_{i=1}^{d'} \left(1 - \left|\Pr(\hat{\mathbf{w}}_i \leq 0 | \sigma_i < 0) - \Pr(\hat{\mathbf{w}}_i \leq 0 | \sigma_i > 0)\right|\right)
\end{aligned}$$

Using Pinsker's inequality and the fact that $\hat{\mathbf{w}}$ is a deterministic function of the training set S, this is at least

$$\frac{1}{2d'}\left(\min\{Yb, B/\sqrt{d'}\}\right)^2 \sum_{i=1}^{d'} \left(1 - \sqrt{\frac{1}{2} D_{kl}\left(p(S|\sigma_i > 0) || p(S|\sigma_i < 0)\right)}\right), \tag{3}$$

where D_{kl} is the Kullback-Leibler (KL) divergence, and p is the probability measure of the sample. Since the training set is composed of m i.i.d. instances, we can use the chain rule

and get that this divergence equals $mD_{kl}\left(p((\mathbf{x},y)|\sigma_i>0)||p((\mathbf{x},y)|\sigma_i<0)\right)$. Moreover, we note that

$$\begin{aligned} p((\mathbf{x},y)|\sigma_i) = &\ p(\mathbf{x}=\mathbf{e}_i)p((\mathbf{x},y)|\sigma_i,\mathbf{x}_i=\mathbf{e}_i)+p(\mathbf{x}\neq\mathbf{e}_i)p((\mathbf{x},y)|\sigma_i,\mathbf{x}\neq\mathbf{e}_i) \\ = &\ \frac{1}{d'}p((\mathbf{x},y)|\sigma_i,\mathbf{x}=\mathbf{e}_i)+\left(1-\frac{1}{d'}\right)p((\mathbf{x},y)|\sigma_i,\mathbf{x}\neq\mathbf{e}_i), \end{aligned}$$

and therefore, by joint convexity of the KL-divergence[3]

$$\begin{aligned} &D_{kl}(p((\mathbf{x},y)|\sigma_i>0)||p(\mathbf{x},y)|\sigma_i<0)) \\ &\qquad \leq \ \frac{1}{d'}D_{kl}\left(p((\mathbf{x},y)|\sigma_i>0,\mathbf{x}=\mathbf{e}_i)||p((\mathbf{x},y)|\sigma_i<0,\mathbf{x}=\mathbf{e}_i)\right) \\ &\qquad\quad +\left(1-\frac{1}{d'}\right)D_{kl}\left(p((\mathbf{x},y)|\sigma_i>0,\mathbf{x}\neq\mathbf{e}_i)||p((\mathbf{x},y)|\sigma_i<0,\mathbf{x}\neq\mathbf{e}_i)\right). \end{aligned}$$

Since the distribution of y is independent of σ_i, conditioned on $\mathbf{x}\neq\mathbf{e}_i$, this equals

$$\frac{1}{d'}D_{kl}\left(p(y|\sigma_i>0,\mathbf{x}=\mathbf{e}_i)||p(y|\sigma_i<0,\mathbf{x}=\mathbf{e}_i)\right). \tag{4}$$

The divergence in this equation is simply the KL divergence between two Bernoulli random variables, one with parameter $\frac{1}{2}(1+b)$, and the other with parameter $\frac{1}{2}(1-b)$. We now use Lemma 4 to upper bound (4) by

$$\frac{b^2}{d'}\left(\frac{1}{\frac{1}{2}(1+b)}+\frac{1}{\frac{1}{2}(1-b)}\right)=\frac{2b^2}{d'}\left(\frac{1}{1+b}+\frac{1}{1-b}\right)\leq\frac{2b^2}{d'}\left(1+\frac{1}{1/2}\right)=\frac{6b^2}{d'},$$

where we used the fact that $b\in[0,1/2]$. Summarizing the discussion so far, we showed that

$$D_{kl}\left(p(S|\sigma_i<0)||p(S|\sigma_i>0)\right) \ = \ m\ D_{kl}\left(p((\mathbf{x},y)|\sigma_i<0)||p((\mathbf{x},y)|\sigma_i>0)\right) \ = \ \frac{6mb^2}{d'}.$$

Plugging this back into (3), we get that the excess risk is lower bounded by

$$\begin{aligned} \frac{1}{2d'}\left(\min\{Yb,B/\sqrt{d'}\}\right)^2\sum_{i=1}^{d'}\left(1-\sqrt{\frac{3mb^2}{d'}}\right) &= \left(\min\{Yb,B/\sqrt{d'}\}\right)^2\frac{1}{2}\left(1-\sqrt{\frac{3mb^2}{d'}}\right) \\ &\geq \left(\min\{Yb,B/\sqrt{d'}\}\right)^2\frac{1}{2}\left(1-\sqrt{\frac{3m(d'/6m)}{d'}}\right) \\ &\geq 0.14\left(\min\{Yb,B/\sqrt{d'}\}\right)^2 \\ &= 0.14\left(\min\left\{Y\min\left\{\frac{1}{2},\sqrt{\frac{d'}{6m}}\right\},\frac{B}{\sqrt{d'}}\right\}\right)^2 \\ &= 0.14\min\left\{\frac{1}{4}Y^2,\frac{d'Y^2}{6m},\frac{B^2}{d'}\right\}. \end{aligned}$$

3. Specifically, we're using the fact that by Jensen's inequality, for any probability distributions p_1,p_2,q_1,q_2 and $\lambda\in[0,1]$, it holds that $D_{KL}((1-\lambda)p_1+\lambda p_2||(1-\lambda)q_1+\lambda q_2)\leq(1-\lambda)D_{KL}(p_1||q_1)+\lambda D_{KL}(p_2||q_2)$. See also Cover and Thomas (2006), theorem 2.7.2.

Now, recall that d' is a free parameter of value at most d. We now distinguish between two cases:

- If $d > \sqrt{6m}B/Y$, then we pick $d' = \lceil \sqrt{6m}B/Y \rceil$, and get that the expression above is at least

$$\begin{aligned} 0.14 \min\left\{\frac{1}{4}Y^2, \frac{B^2}{d'}\right\} &\geq 0.14 \min\left\{\frac{1}{4}Y^2, \frac{B^2}{\max\left\{1, 2\sqrt{6m}\frac{B}{Y}\right\}}\right\} \\ &= 0.14 \min\left\{\frac{1}{4}Y^2, B^2, \frac{BY}{2\sqrt{6m}}\right\}. \end{aligned}$$

- If $d \leq \sqrt{6m}B/Y$, we pick $d' = d$, and note that $\frac{d'Y^2}{6m} \leq \frac{B^2}{d}$ in this case. Therefore, the expression above is at least

$$0.14 \min\left\{\frac{1}{4}Y^2, \frac{dY^2}{6m}\right\}$$

Combining the two cases, we get that a lower bound of the form

$$c \ \min\left\{Y^2, B^2, \frac{dY^2}{m}, \frac{BY}{\sqrt{m}}\right\},$$

where c is a universal constant. ■

With Thm. 3 and Thm. 5 at hand, we now turn to prove our main result:

Proof [Proof of Thm. 1] Taking the maximum of Thm. 3 and Thm. 5, and using the fact that $B \geq 2Y$, we get a lower bound of

$$c \max\left\{\min\left\{Y^2, \frac{B^2}{m}\right\} \ , \ \min\left\{Y^2, \frac{dY^2}{m}, \frac{BY}{\sqrt{m}}\right\}\right\}$$

for some constant c. If $m \leq (B^2/Y^2)$, this is at least Y^2, and otherwise it is

$$\begin{aligned} c \max\left\{\frac{B^2}{m} \ , \ \min\left\{\frac{dY^2}{m}, \frac{BY}{\sqrt{m}}\right\}\right\} &\geq \frac{c}{2}\left(\frac{B^2}{m} + \min\left\{\frac{dY^2}{m}, \frac{BY}{\sqrt{m}}\right\}\right) \\ &\geq \frac{c}{2}\min\left\{\frac{B^2 + dY^2}{m}, \frac{BY}{\sqrt{m}}\right\}. \end{aligned}$$

Combining the two cases, the result follows. ■

Acknowledgments

We thank Nati Srebro and the anonymous reviewers for several helpful comments. This research is partially supported by an Israeli Science Foundation grant (no. 425/13) and an FP7 Marie Curie CIG grant.

References

M. Anthony and P. Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, 1999.

J.-Y. Audibert and O. Catoni. Robust linear least squares regression. *The Annals of Statistics*, 39(5):2766–2794, 2011.

K. Azoury and M. Warmuth. Relative loss bounds for on-line density estimation with the exponential family of distributions. *Machine Learning*, 43(3):211–246, 2001.

P. Bartlett and S. Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *The Journal of Machine Learning Research*, 3:463–482, 2003.

T. Cover and J. Thomas. *Elements of Information Theory*. John Wiley & Sons, 2nd edition, 2006.

A. Gibbs and F. Su. On choosing and bounding probability metrics. *International Statistical Review*, 70(3):419–435, 2002.

D. Hsu, S. M Kakade, and T. Zhang. Random design analysis of ridge regression. *Foundations of Computational Mathematics*, 14(3):569–600, 2014.

V. Koltchinskii. *Oracle Inequalities in Empirical Risk Minimization and Sparse Recovery Problems: Ecole d'Eté de Probabilités de Saint-Flour XXXVIII-2008*, volume 2033. Springer, 2011.

G. Lecué and S. Mendelson. Performance of empirical risk minimization in linear aggregation. *arXiv Preprint arXiv:1402.5763*, 2014.

W. Lee, P. Bartlett, and R. Williamson. The importance of convexity in learning with squared loss. *Information Theory, IEEE Transactions on*, 44(5):1974–1980, 1998.

S. Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2012.

A. Singer, S. Kozat, and M. Feder. Universal linear least squares prediction: upper and lower bounds. *Information Theory, IEEE Transactions on*, 48(8):2354–2362, 2002.

N. Srebro, K. Sridharan, and A. Tewari. Smoothness, low noise and fast rates. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2199–2207, 2010.

A. Tsybakov. Optimal rates of aggregation. In *Learning Theory and Kernel Machines*, pages 303–313. Springer, 2003.

V. Vovk. Competitive on-line statistics. *International Statistical Review*, 69(2):213–248, 2001.

T. Zhang. Learning bounds for kernel regression using effective data dimensionality. *Neural Computation*, 17(9):2077–2098, 2005.

Journal of Machine Learning Research 16 (2015) 3487-3602 Submitted 10/14; Published 12/15

Minimax Analysis of Active Learning

Steve Hanneke STEVE.HANNEKE@GMAIL.COM
Princeton, NJ 08542
Liu Yang YANGLI@US.IBM.COM
IBM T. J. Watson Research Center, Yorktown Heights, NY 10598

Editor: Alexander Rakhlin

Abstract

This work establishes distribution-free upper and lower bounds on the minimax label complexity of active learning with general hypothesis classes, under various noise models. The results reveal a number of surprising facts. In particular, under the noise model of Tsybakov (2004), the minimax label complexity of active learning with a VC class is always asymptotically smaller than that of passive learning, and is typically significantly smaller than the best previously-published upper bounds in the active learning literature. In high-noise regimes, it turns out that all active learning problems of a given VC dimension have roughly the same minimax label complexity, which contrasts with well-known results for bounded noise. In low-noise regimes, we find that the label complexity is well-characterized by a simple combinatorial complexity measure we call the *star number*. Interestingly, we find that almost all of the complexity measures previously explored in the active learning literature have worst-case values exactly equal to the star number. We also propose new active learning strategies that nearly achieve these minimax label complexities.

Keywords: active learning, selective sampling, sequential design, adaptive sampling, statistical learning theory, margin condition, Tsybakov noise, sample complexity, minimax analysis

1. Introduction

In many machine learning applications, in the process of training a high-accuracy classifier, the primary bottleneck in time and effort is often the annotation of the large quantities of data required for supervised learning. Active learning is a protocol designed to reduce this cost by allowing the learning algorithm to sequentially identify highly-informative data points to be annotated. In the specific protocol we study below, called *pool-based* active learning, the learning algorithm is initially given access to a large pool of unlabeled data points, which are considered inexpensive and abundant. It is then able to select any unlabeled data point from the pool and request its label. Given the label of this point, the algorithm can then select another unlabeled data point to be labeled, and so on. This interactive process continues for some prespecified number of rounds, after which the algorithm must halt and produce a classifier. This contrasts with *passive learning*, where the data points to be labeled are chosen at *random*. The hope is that, by sequentially selecting the data points to be labeled, the active learning algorithm can direct the annotation effort toward only the highly-informative data points, given the information already gathered from previously-labeled data, and thereby reduce the total number of labels required to produce

a classifier capable of predicting the labels of new instances with a desired level of accuracy. This active learning protocol has been used in practice for a variety of learning problems, often with significant reductions in the time and effort required for data annotation (see Settles, 2012, for a survey of several such applications).

This article studies the theoretical capabilities of active learning, regarding the number of label requests sufficient to learn a classifier to a desired error rate, known as the *label complexity*. There is now a substantial literature on this subject (see Hanneke, 2014, for a survey of known results), but on the important question of *optimal* performance in the general setting, the gaps present in the literature are quite substantial in some cases. In this work, we address this question by carefully studying the *minimax* performance. Specifically, we are interested in the *minimax label complexity*, defined as the smallest (over the choice of active learning algorithm) worst-case number of label requests sufficient for the active learning algorithm to produce a classifier of a specified error rate, in the context of various noise models (e.g., Tsybakov noise, bounded noise, agnostic noise, etc.). We derive upper and lower bounds on the minimax label complexity for several noise models, which reveal a variety of interesting and (in some cases) surprising observations. Furthermore, in establishing the upper bounds, we propose a novel active learning strategy, which often achieves significantly smaller label complexities than the active learning methods studied in the prior literature.

1.1 The Prior Literature on the Theory of Active Learning

Before getting into the technical details, we first review some background information about the prior literature on the theory of active learning. This will also allow us to introduce the key contributions of the present work.

The literature on the theory of active learning began with studies of the *realizable case*, a setting in which the labels are assumed to be consistent with some classifier in a known hypothesis class, and have no noise (Cohn, Atlas, and Ladner, 1994; Freund, Seung, Shamir, and Tishby, 1997; Dasgupta, 2004, 2005). In this simple setting, Dasgupta (2005) supplied the first general analysis of the label complexity of active learning, applicable to arbitrary hypothesis classes. However, Dasgupta (2005) found that there are a range of minimax label complexities, depending on the structure of the hypothesis class, so that even among hypothesis classes of roughly the same minimax sample complexities for passive learning, there can be widely varying minimax label complexities for active learning. In particular, he found that some hypothesis classes (e.g., interval classifiers) have minimax label complexity essentially no better than that of passive learning, while others have a minimax label complexity exponentially smaller than that of passive learning (e.g., threshold classifiers). Furthermore, most nontrivial hypothesis classes of interest in learning theory seem to fall into the former category, with minimax label complexities essentially no better than passive learning. Fortunately, Dasgupta (2005) also found that in some of these hard cases, it is still possible to show improvements over passive learning under restrictions on the data distribution.

Stemming from these observations, much of the literature on active learning in the realizable case has focused on describing various special conditions under which the label complexity of active learning is significantly better than that of passive learning: for instance,

by placing restrictions on the marginal distribution of the unlabeled data (e.g., Dasgupta, Kalai, and Monteleoni, 2005; Balcan, Broder, and Zhang, 2007; El-Yaniv and Wiener, 2012; Balcan and Long, 2013; Hanneke, 2014), or abandoning the minimax approach by expressing the label complexity with an explicit dependence on the optimal classifier (e.g., Dasgupta, 2005; Balcan, Hanneke, and Vaughan, 2010; Hanneke, 2009b, 2012). In the general case, such results have been abstracted into various distribution-dependent (or sometimes data-dependent) *complexity measures*, such as the *splitting index* (Dasgupta, 2005), the *disagreement coefficient* (Hanneke, 2007b, 2009b), the *extended teaching dimension growth function* (Hanneke, 2007a), and the related *version space compression set size* (El-Yaniv and Wiener, 2010, 2012; Wiener, Hanneke, and El-Yaniv, 2015). For each of these, there are general upper bounds (and in some cases, minimax lower bounds) on the label complexities achievable by active learning methods in the realizable case, expressed in terms of the complexity measure. By expressing bounds on the label complexity in terms of these quantities, the analysis of label complexities achievable by active learning in the realizable case has been effectively reduced to the problem of bounding one of these complexity measures. In particular, these complexity measures are capable of exhibiting a range of behaviors, corresponding to the range of label complexities achievable by active learning. For certain values of the complexity measures, the resulting bounds reveal significant improvements over the minimax sample complexity of passive learning, while for other values, the resulting bounds are essentially no better than the minimax sample complexity of passive learning.

Moving beyond these initial studies of the realizable case, the more-recent literature has developed active learning algorithms that are provably robust to label noise. This advance was initiated by the seminal work of Balcan, Beygelzimer, and Langford (2006, 2009) on the A^2 (Agnostic Active) algorithm, and continued by a number of subsequent works (e.g., Dasgupta, Hsu, and Monteleoni, 2007; Balcan, Broder, and Zhang, 2007; Castro and Nowak, 2006, 2008; Hanneke, 2007a, 2009a,b, 2011, 2012; Minsker, 2012; Koltchinskii, 2010; Beygelzimer, Dasgupta, and Langford, 2009; Beygelzimer, Hsu, Langford, and Zhang, 2010; Hsu, 2010; Ailon, Begleiter, and Ezra, 2012; Hanneke and Yang, 2012). When moving into the analysis of label complexity in noisy settings, the literature continues to follow the same intuition from the realizable case: that is, that there should be some active learning problems that are inherently hard, sometimes no better than passive learning, while others are significantly easier, with significant savings compared to passive learning. As such, the general label complexity bounds proven in noisy settings have tended to follow similar patterns to those found in the realizable case. In some scenarios, the bounds reflect interesting savings compared to passive learning, while in other scenarios the bounds do not reflect any improvements at all. However, unlike the realizable case, these upper bounds on the label complexities of the various proposed methods for noisy settings lacked complementary minimax lower bounds showing that they were accurately describing the fundamental capabilities of active learning in these settings. For instance, in the setting of Tsybakov noise, there are essentially only two types of general lower bounds on the minimax label complexity in the prior literature: (1) lower bounds that hold for all nontrivial hypothesis classes of a given VC dimension, which therefore reflect a kind of best-case scenario (Hanneke, 2011, 2014), and (2) lower bounds inherited from the realizable case (which is a special case of Tsybakov noise). In particular, both of these lower bounds are always smaller than the minimax sample complexity of passive learning under Tsybakov noise. Thus, although the

upper bounds on the label complexity of active learning in the literature are sometimes no better than the minimax sample complexity of passive learning, the existing lower bounds are unable to confirm that active learning truly cannot outperform passive learning in these scenarios. This gap in our understanding of active learning with noise has persisted for a number of years now, without really receiving a good explanation for why the gap exists and how it might be closed.

In the present work, we show that there is a very good reason for why better lower bounds have not been discovered in general for the noisy case. For certain ranges of the noise parameters (corresponding to the high-noise regime), these simple lower bounds are actually *tight* (up to certain constant and logarithmic factors): that is, the upper bounds can actually be reduced to nearly match these basic lower bounds. Proving this surprising fact requires the introduction of a new type of active learning strategy, which selects its queries based on both the structure of the hypothesis class and the estimated variances of the labels. In particular, in these high-noise regimes, we find that *all* hypothesis classes of the same VC dimension have essentially the same minimax label complexities (up to logarithmic factors), in stark contrast to the well-known differentiation of hypothesis classes observed in the realizable case by Dasgupta (2005).

For the remaining range of the noise parameters (the low-noise regime), we argue that the label complexity takes a value sometimes larger than this basic lower bound, yet still typically smaller than the known upper bounds. In this case, we further argue that the minimax label complexity is well-characterized by a simple combinatorial complexity measure, which we call the *star number*. In particular, these results reveal that for nonextremal parameter values, the minimax label complexity of active learning under Tsybakov noise with *any* VC class is *always* smaller than that of passive learning, a fact not implied by any results in the prior literature.

We further find that the star number can be used to characterize the minimax label complexities for a variety of other noise models. Interestingly, we also show that almost all of the distribution-dependent or data-dependent complexity measures from the prior literature on the label complexity of active learning are exactly *equal* to the star number when maximized over the choice of distribution or data set (including all of those mentioned above). Thus, the star number represents a unifying core concept within these disparate styles of analysis.

1.2 Our Contributions

We summarize a few of the main contributions and interesting implications of this work.

- We develop a general noise-robust active learning strategy, which unlike previously-proposed general methods, selects its queries based on both the structure of the hypothesis class *and* the estimated variances of the labels.

- We obtain the first near-matching general distribution-free upper and lower bounds on the minimax label complexity of active learning, under a variety of noise models.

- In many cases, the upper bounds significantly improve over the best upper bounds implied by the prior literature.

- The upper bounds for Tsybakov noise *always* reflect improvements over the minimax sample complexity of passive learning (for non-extremal noise parameter values), a feat not previously known to be possible.
- In high-noise regimes of Tsybakov noise, our results imply that all hypothesis classes of a given VC dimension have roughly the same minimax label complexity (up to logarithmic factors), in contrast to well-known results for bounded noise. This fact is not implied by any results in the prior literature.
- We express our upper and lower bounds on the label complexity in terms of a simple combinatorial complexity measure, which we refer to as the *star number*.
- We show that for any hypothesis class, almost every complexity measure proposed to date in the active learning literature has worst-case value exactly *equal* to the star number, thus unifying the disparate styles of analysis in the active learning literature. We also prove that the doubling dimension is bounded if and only if the star number is finite.
- For most of the noise models studied here, we exhibit examples of hypothesis classes spanning the gaps between the upper and lower bounds, thus demonstrating that the gaps cannot generally be reduced (aside from logarithmic factors) without introducing additional complexity measures.
- We prove a separation result for Tsybakov noise vs the Bernstein class condition, establishing that the respective minimax label complexities can be significantly different. This contrasts with passive learning, where they are known to be equivalent up to a logarithmic factor.

The algorithmic techniques underlying the proofs of the most-interesting of our upper bounds involve a combination of the disagreement-based strategy of Cohn, Atlas, and Ladner (1994) (and the analysis thereof by Hanneke, 2011, and Wiener, Hanneke, and El-Yaniv, 2015), along with a repeated-querying technique of Kääriäinen (2006), modified to account for variations in label variances so that the algorithm does not waste too many queries determining the optimal classification of highly-noisy points; this modification represents the main algorithmic innovation in this work. In a supporting role, we also rely on auxiliary lemmas on the construction of ε-nets and ε-covers based on random samples, and the use of these to effectively discretize the instance space. The mathematical techniques underlying the proofs of the lower bounds are largely taken directly from the literature. Most of the lower bounds are established by a combination of a technique originating with Kääriäinen (2006) and refined by Beygelzimer, Dasgupta, and Langford (2009) and Hanneke (2011, 2014), and a technique of Raginsky and Rakhlin (2011) for incorporating a complexity measure into the lower bounds.

We note that, while the present work focuses on the distribution-free setting, in which the marginal distribution over the instance space is unrestricted, our results reveal that low-noise settings can still benefit from distribution-dependent analysis, as expected given the aforementioned observations by Dasgupta (2005) for the realizable case. For instance, under Tsybakov noise, it is often possible to obtain stronger upper bounds in low-noise

regimes under assumptions restricting the distribution of the unlabeled data (see e.g., Balcan, Broder, and Zhang, 2007). We leave for future work the important problem of characterizing the minimax label complexity of active learning in the general case for an arbitrary fixed marginal distribution over the instance space.

1.3 Outline

The rest of this article is organized as follows. Section 2 introduces the formal setting and basic notation used throughout, followed in Section 3 with the introduction of the noise models studied in this work. Section 4 defines a combinatorial complexity measure – the star number – in terms of which we will express the label complexity bounds below. Section 5 provides statements of the main results of this work: upper and lower bounds on the minimax label complexities of active learning under each of the noise models defined in Section 3. That section also includes a discussion of the results, and a brief sketch of the arguments underlying the most-interesting among them. Section 6 compares the results from Section 5 to the known results on the minimax sample complexity of passive learning, revealing which scenarios yield improvements of active over passive. Next, in Section 7, we go through the various results on the label complexity of active learning from the literature, along with their corresponding complexity measures (most of which are distribution-dependent or data-dependent). We argue that all of these complexity measures are exactly equal to the star number when maximized over the choice of distribution or data set. This section also relates the star number to the well-known concept of *doubling dimension*, in particular showing that the doubling dimension is bounded if and only if the star number is finite.

We note that the article is written with the intention that it be read in-order; for instance, while Appendix B contains proofs of the results in Section 5, those proofs refer to quantities and results introduced in Sections 6 and 7 (which follow Section 5, but precede Appendix B).

2. Definitions

The rest of this paper makes use of the following formal definitions. There is a space $\mathcal{X}$, called the *instance space*. We suppose $\mathcal{X}$ is equipped with a σ-algebra $\mathcal{B}_{\mathcal{X}}$, and for simplicity we will assume $\{\{x\} : x \in \mathcal{X}\} \subseteq \mathcal{B}_{\mathcal{X}}$. There is also a set $\mathcal{Y} = \{-1, +1\}$, known as the *label space*. Any measurable function $h : \mathcal{X} \to \mathcal{Y}$ is called a *classifier*. There is an arbitrary set $\mathbb{C}$ of classifiers, known as the *hypothesis class*. To focus on nontrivial cases, we suppose $|\mathbb{C}| \geq 3$ throughout.

For any probability measure P over $\mathcal{X} \times \mathcal{Y}$ and any $x \in \mathcal{X}$, define $\eta(x; P) = \mathbb{P}(Y = +1 | X = x)$ for $(X, Y) \sim P$, and let $f_P^\star(x) = \text{sign}(2\eta(x; P) - 1)$ denote the *Bayes optimal classifier*,[1] where $\text{sign}(t) = +1$ if $t \geq 0$, and $\text{sign}(t) = -1$ if $t < 0$. Define the *error rate* of a classifier h with respect to P as $\text{er}_P(h) = P((x, y) : h(x) \neq y)$.

1. Since conditional probabilities are only defined up to probability zero differences, there can be multiple valid functions $\eta(\cdot; P)$ and $f_P^\star$, with any two such functions being equal with probability one. As such, we will interpret statements such as "$f_P^\star \in \mathbb{C}$" to mean that *there exists* a version of $f_P^\star$ contained in $\mathbb{C}$, and similarly for other claims and conditions for $f_P^\star$ and $\eta(\cdot; P)$.

In the learning problem, there is a *target distribution* $\mathcal{P}_{XY}$ over $\mathcal{X} \times \mathcal{Y}$, and a *data sequence* $(X_1, Y_1), (X_2, Y_2), \ldots$, which are independent $\mathcal{P}_{XY}$-distributed random variables. However, in the active learning protocol, the Y_i values are initially "hidden" until individually requested by the algorithm (see below). We refer to the sequence $X_1, X_2, \ldots$ as the *unlabeled data sequence*.[2] We will sometimes denote by $\mathcal{P}$ the marginal distribution of $\mathcal{P}_{XY}$ over $\mathcal{X}$: that is, $\mathcal{P}(\cdot) = \mathcal{P}_{XY}(\cdot \times \mathcal{Y})$.

In the *pool-based active learning* protocol,[3] we define an *active learning algorithm* $\mathcal{A}$ as an algorithm taking as input a budget $n \in \mathbb{N} \cup \{0\}$, and proceeding as follows. The algorithm initially has access to the unlabeled data sequence $X_1, X_2, \ldots$. If $n > 0$, the algorithm may then select an index $i_1 \in \mathbb{N}$ and request to observe the label Y_{i_1}. The algorithm may then observe the value of Y_{i_1}, and if $n \geq 2$, then based on both the unlabeled sequence and this new observation Y_{i_1}, it may select another index $i_2 \in \mathbb{N}$ and request to observe Y_{i_2}. This continues for a number of rounds at most n (i.e., it may request at most n labels), after which the algorithm must halt and produce a classifier $\hat{h}_n$. More formally, an active learning algorithm is defined by a random sequence $\{i_t\}_{t=1}^{\infty}$ in $\mathbb{N}$, a random variable N in $\mathbb{N}$, and a random classifier $\hat{h}_n$, satisfying the following properties. Each i_t is conditionally independent from $\{(X_i, Y_i)\}_{i=1}^{\infty}$ given $\{i_j\}_{j=1}^{t-1}$, $\{Y_{i_j}\}_{j=1}^{t-1}$, and $\{X_i\}_{i=1}^{\infty}$. The random variable N always has $N \leq n$, and for any $k \in \{0, \ldots, n\}$, $\mathbb{1}[N = k]$ is independent from $\{(X_i, Y_i)\}_{i=1}^{\infty}$ given $\{i_j\}_{j=1}^{k}$, $\{Y_{i_j}\}_{j=1}^{k}$, and $\{X_i\}_{i=1}^{\infty}$. Finally, $\hat{h}_n$ is independent from $\{(X_i, Y_i)\}_{i=1}^{\infty}$ given N, $\{i_j\}_{j=1}^{N}$, $\{Y_{i_j}\}_{j=1}^{N}$, and $\{X_i\}_{i=1}^{\infty}$.

We are now ready for the definition of our primary quantity of study: the minimax label complexity. In the next section, we define several well-known noise models as specifications of the set $\mathbb{D}$ referenced in this definition.

Definition 1 *For a given set $\mathbb{D}$ of probability measures on $\mathcal{X} \times \mathcal{Y}$, $\forall \varepsilon \geq 0$, $\forall \delta \in [0, 1]$, the* minimax label complexity *(of active learning) under $\mathbb{D}$ with respect to $\mathbb{C}$, denoted $\Lambda_{\mathbb{D}}(\varepsilon, \delta)$, is the smallest $n \in \mathbb{N} \cup \{0\}$ such that there exists an active learning algorithm $\mathcal{A}$ with the property that, for every $\mathcal{P}_{XY} \in \mathbb{D}$, the classifier $\hat{h}_n$ produced by $\mathcal{A}(n)$ based on the (independent $\mathcal{P}_{XY}$-distributed) data sequence $(X_1, Y_1), (X_2, Y_2), \ldots$ satisfies*

$$\mathbb{P}\left(\mathrm{er}_{\mathcal{P}_{XY}}\left(\hat{h}_n\right) - \inf_{h \in \mathbb{C}} \mathrm{er}_{\mathcal{P}_{XY}}(h) > \varepsilon\right) \leq \delta.$$

If no such n exists, we define $\Lambda_{\mathbb{D}}(\varepsilon, \delta) = \infty$.

Following Vapnik and Chervonenkis (1971); Anthony and Bartlett (1999), we say a collection of sets $\mathcal{T} \subseteq 2^{\mathcal{X}}$ *shatters* a sequence $S \in \mathcal{X}^k$ (for $k \in \mathbb{N}$) if $\{A \cap S : A \in \mathcal{T}\} = 2^S$.

2. Although, in practice, we would expect to have access to only a finite number of unlabeled samples, we expect this number would often be quite large (as unlabeled samples are considered inexpensive and abundant in many applications). For simplicity, and to focus the analysis purely on the number of *labels* required for learning, we approximate this scenario by supposing an *inexhaustible* source of unlabeled samples. We leave open the question of the number of unlabeled samples sufficient to obtain the minimax label complexity; in particular, we expect the number of such samples used by the methods obtaining our upper bounds to be quite large indeed.
3. Although technically we study the pool-based active learning protocol, all of our results apply equally well to the stream-based (selective sampling) model of active learning (in which the algorithm must decide whether or not to request the label Y_i before observing any X_j with $j > i$ or requesting any Y_j with $j > i$).

The *VC dimension* of $\mathcal{T}$ is then defined as the largest $k \in \mathbb{N} \cup \{0\}$ such that there exists $S \in \mathcal{X}^k$ shattered by $\mathcal{T}$; if no such largest k exists, the VC dimension is defined to be ∞. Overloading this terminology, the VC dimension of a set $\mathcal{H}$ of classifiers is defined as the VC dimension of the collection of sets $\{\{x : h(x) = +1\} : h \in \mathcal{H}\}$. Throughout this article, we denote by d the VC dimension of $\mathbb{C}$. We are particularly interested in the case $d < \infty$, in which case $\mathbb{C}$ is called a *VC class*.

For any set $\mathcal{H}$ of classifiers, define $\mathrm{DIS}(\mathcal{H}) = \{x \in \mathcal{X} : \exists h, g \in \mathcal{H} \text{ s.t. } h(x) \neq g(x)\}$, the *region of disagreement* of $\mathcal{H}$. Also, for any classifier h, any $r \geq 0$, and any probability measure P on $\mathcal{X}$, define $\mathrm{B}_P(h, r) = \{g \in \mathbb{C} : P(x : g(x) \neq h(x)) \leq r\}$, the *$r$-ball centered at h*.

Before proceeding, we introduce a few additional notational conventions that help to simplify the theorem statements and proofs. For any $\mathbb{R}$-valued functions f and g, we write $f(x) \lesssim g(x)$ (or equivalently $g(x) \gtrsim f(x)$) to express the fact that there is a *universal* finite numerical constant $c > 0$ such that $f(x) \leq cg(x)$. For any $x \in [0, \infty]$, we define $\mathrm{Log}(x) = \max\{\ln(x), 1\}$, where $\ln(0) = -\infty$ and $\ln(\infty) = \infty$. For simplicity, we define $\frac{\infty}{\mathrm{Log}(\infty)} = \infty$, but in any other context, we always define $0 \cdot \infty = 0$, and also define $\frac{a}{0} = \infty$ for any $a > 0$. For any function $\phi : \mathbb{R} \to \mathbb{R}$, we use the notation "$\lim_{\gamma \to 0} \phi(\gamma)$" to indicating taking the limit as γ approaches 0 *from above*: i.e., $\gamma \downarrow 0$. For $a, b \in \mathbb{R}$, we denote $a \wedge b = \min\{a, b\}$ and $a \vee b = \max\{a, b\}$. Finally, we remark that some of the claims below technically require additional qualifications to guarantee measurability of certain quantities (as is typically the case in empirical process theory); see Blumer, Ehrenfeucht, Haussler, and Warmuth (1989); van der Vaart and Wellner (1996, 2011) for some discussion of this issue. For simplicity, we do not mention these issues in the analysis below; rather, we implicitly qualify all of these results with the condition that $\mathbb{C}$ is such that all of the random variables and events arising in the proofs are measurable.

3. Noise Models

We now introduce the noise models under which we will study the minimax label complexity of active learning. These are defined as sets of probability measures on $\mathcal{X} \times \mathcal{Y}$, corresponding to specifications of the set $\mathbb{D}$ in Definition 1.

- (Realizable Case) Define RE as the collection of $\mathcal{P}_{XY}$ for which $f^\star_{\mathcal{P}_{XY}} \in \mathbb{C}$ and $2\eta(\cdot; \mathcal{P}_{XY}) - 1 = f^\star_{\mathcal{P}_{XY}}(\cdot)$ (almost everywhere w.r.t. $\mathcal{P}$).

- (Bounded Noise) For $\beta \in [0, 1/2)$, define $\mathrm{BN}(\beta)$ as the collection of joint distributions $\mathcal{P}_{XY}$ over $\mathcal{X} \times \mathcal{Y}$ such that $f^\star_{\mathcal{P}_{XY}} \in \mathbb{C}$ and

$$\mathcal{P}\left(x : |\eta(x; \mathcal{P}_{XY}) - 1/2| \geq 1/2 - \beta\right) = 1.$$

- (Tsybakov Noise) For $a \in [1, \infty)$ and $\alpha \in (0, 1)$, define $\mathrm{TN}(a, \alpha)$ as the collection of joint distributions $\mathcal{P}_{XY}$ over $\mathcal{X} \times \mathcal{Y}$ such that $f^\star_{\mathcal{P}_{XY}} \in \mathbb{C}$ and $\forall \gamma > 0$,

$$\mathcal{P}\left(x : |\eta(x; \mathcal{P}_{XY}) - 1/2| \leq \gamma\right) \leq a' \gamma^{\alpha/(1-\alpha)},$$

  where $a' = (1 - \alpha)(2\alpha)^{\alpha/(1-\alpha)} a^{1/(1-\alpha)}$.

- (Bernstein Class Condition) For $a \in [1, \infty)$ and $\alpha \in [0, 1]$, define $\mathrm{BC}(a, \alpha)$ as the collection of joint distributions $\mathcal{P}_{XY}$ over $\mathcal{X} \times \mathcal{Y}$ such that, $\exists h_{\mathcal{P}_{XY}} \in \mathbb{C}$ for which $\forall h \in \mathbb{C}$,
$$\mathcal{P}(x : h(x) \neq h_{\mathcal{P}_{XY}}(x)) \leq a(\mathrm{er}_{\mathcal{P}_{XY}}(h) - \mathrm{er}_{\mathcal{P}_{XY}}(h_{\mathcal{P}_{XY}}))^{\alpha}.$$
- (Benign Noise) For $\nu \in [0, 1/2]$, define $\mathrm{BE}(\nu)$ as the collection of all joint distributions $\mathcal{P}_{XY}$ over $\mathcal{X} \times \mathcal{Y}$ such that $f^{\star}_{\mathcal{P}_{XY}} \in \mathbb{C}$ and $\mathrm{er}_{\mathcal{P}_{XY}}(f^{\star}_{\mathcal{P}_{XY}}) \leq \nu$.
- (Agnostic Noise) For $\nu \in [0, 1]$, define $\mathrm{AG}(\nu)$ as the collection of all joint distributions $\mathcal{P}_{XY}$ over $\mathcal{X} \times \mathcal{Y}$ such that $\inf_{h \in \mathbb{C}} \mathrm{er}_{\mathcal{P}_{XY}}(h) \leq \nu$.

It is known that $\mathrm{RE} \subseteq \mathrm{BN}(\beta) \subseteq \mathrm{BC}(1/(1-2\beta), 1)$, and also that $\mathrm{RE} \subseteq \mathrm{TN}(a, \alpha) \subseteq \mathrm{BC}(a, \alpha)$. Furthermore, $\mathrm{TN}(a, \alpha)$ is equivalent to the conditions in $\mathrm{BC}(a, \alpha)$ being satisfied for *all* classifiers h, rather than merely those in $\mathbb{C}$ (Mammen and Tsybakov, 1999; Tsybakov, 2004; Boucheron, Bousquet, and Lugosi, 2005). All of RE, $\mathrm{BN}(\beta)$, and $\mathrm{TN}(a, \alpha)$ are contained in $\bigcup_{\nu < 1/2} \mathrm{BE}(\nu)$, and in particular, $\mathrm{BN}(\beta) \subseteq \mathrm{BE}(\beta)$.

The realizable case is the simplest setting studied here, corresponding to the "optimistic case" of Vapnik (1998) or the PAC model of Valiant (1984). The bounded noise model has been studied under various names (e.g., Massart and Nédélec, 2006; Giné and Koltchinskii, 2006; Kääriäinen, 2006; Koltchinskii, 2010; Raginsky and Rakhlin, 2011); it is sometimes referred to as *Massart's noise condition*. The Tsybakov noise condition was introduced by Mammen and Tsybakov (1999) in a slightly stronger form (in the related context of discrimination analysis) and was distilled into the form stated above by Tsybakov (2004). There is now a substantial literature on the label complexity under this condition, both for passive learning and active learning (e.g., Mammen and Tsybakov, 1999; Tsybakov, 2004; Bartlett, Jordan, and McAuliffe, 2006; Koltchinskii, 2006; Balcan, Broder, and Zhang, 2007; Hanneke, 2011, 2012, 2014; Hanneke and Yang, 2012). However, in much of this literature, the results are in fact established under the weaker assumption given by the Bernstein class condition (Bartlett, Mendelson, and Philips, 2004), which is known to be implied by the Tsybakov noise condition (Mammen and Tsybakov, 1999; Tsybakov, 2004). For passive learning, it is known that the minimax sample complexities under Tsybakov noise and under the Bernstein class condition are equivalent up to a logarithmic factor. Interestingly, our results below imply that this is not the case for active learning. The benign noise condition (studied by Hanneke, 2009b) requires only that the Bayes optimal classifier be contained within the hypothesis class, and that the Bayes error rate be at most the value of the parameter ν. The agnostic noise condition (sometimes called *adversarial noise* in related contexts) is the weakest of the noise assumptions studied here, and admits any distribution for which the best error rate among classifiers in the hypothesis class is at most the value of the parameter ν. This model has been widely studied in the literature, for both passive and active learning (e.g., Vapnik and Chervonenkis, 1971; Vapnik, 1982, 1998; Kearns, Schapire, and Sellie, 1994; Kalai, Klivans, Mansour, and Servedio, 2005; Balcan, Beygelzimer, and Langford, 2006; Hanneke, 2007b,a; Awasthi, Balcan, and Long, 2014).

4. A Combinatorial Complexity Measure

There is presently a substantial literature on distribution-dependent bounds on the label complexities of various active learning algorithms. These bounds are expressed in terms of a

variety of interesting complexity measures, designed to capture the behavior of each of these particular algorithms. These measures of complexity include the disagreement coefficient (Hanneke, 2007b), the reciprocal of the splitting index (Dasgupta, 2005), the extended teaching dimension growth function (Hanneke, 2007a), and the version space compression set size (El-Yaniv and Wiener, 2010, 2012). These quantities have been studied and bounded for a variety of learning problems (see Hanneke, 2014, for a summary). They each have many interesting properties, and in general can exhibit a wide variety of behaviors, as functions of the distribution over $\mathcal{X}$ (and in some cases, the distribution over $\mathcal{X} \times \mathcal{Y}$) and ε, or in some cases, the data itself. However, something remarkable happens when we maximize each of these complexity measures over the choice of distribution (or data set): they all become equal to a simple and easy-to-calculate combinatorial quantity (see Section 7 for proofs of these equivalences). Specifically, consider the following definition.[4]

Definition 2 *Define the* star number $\mathfrak{s}$ *as the largest integer s such that there exist distinct points $x_1, \ldots, x_s \in \mathcal{X}$ and classifiers $h_0, h_1, \ldots, h_s \in \mathbb{C}$ with the property that $\forall i \in \{1, \ldots, s\}$, $\mathrm{DIS}(\{h_0, h_i\}) \cap \{x_1, \ldots, x_s\} = \{x_i\}$; if no such largest integer exists, define $\mathfrak{s} = \infty$.*

For any set $\mathcal{H}$ of functions $\mathcal{X} \to \mathcal{Y}$, any $t \in \mathbb{N}$, $x_1, \ldots, x_t \in \mathcal{X}$, and $h_0, h_1, \ldots, h_t \in \mathcal{H}$, we will say $\{x_1, \ldots, x_t\}$ is a *star set* for $\mathcal{H}$, *witnessed by* $\{h_0, h_1, \ldots, h_t\}$, if $\forall i \in \{1, \ldots, t\}$, $\mathrm{DIS}(\{h_0, h_i\}) \cap \{x_1, \ldots, x_t\} = \{x_i\}$. For brevity, in some instances below, we may simply say that $\{x_1, \ldots, x_t\}$ *is a star set for* $\mathcal{H}$, indicating that $\exists h_0, h_1, \ldots, h_t \in \mathcal{H}$ such that $\{x_1, \ldots, x_t\}$ is a star set for $\mathcal{H}$, witnessed by $\{h_0, h_1, \ldots, h_t\}$. We may also say that $\{x_1, \ldots, x_t\}$ *is a star set for $\mathcal{H}$ centered at $h_0 \in \mathcal{H}$* if $\exists h_1, \ldots, h_t \in \mathcal{H}$ such that $\{x_1, \ldots, x_t\}$ is a star set for $\mathcal{H}$, witnessed by $\{h_0, h_1, \ldots, h_t\}$. For completeness, we also say that $\{\}$ (the empty sequence) is a star set for $\mathcal{H}$ (witnessed by $\{h_0\}$ for any $h_0 \in \mathcal{H}$), for any nonempty $\mathcal{H}$. In these terms, the star number of $\mathbb{C}$ is the maximum possible cardinality of a star set for $\mathbb{C}$, or ∞ if no such maximum exists.

The star number can equivalently be described as the maximum possible degree in the data-induced one-inclusion graph for $\mathbb{C}$ (see Haussler, Littlestone, and Warmuth, 1994), where the maximum is over all possible data sets and nodes in the graph.[5] To relate this to the VC dimension, one can show that the VC dimension is the maximum possible degree of a *hypercube* in the data-induced one-inclusion graph for $\mathbb{C}$ (maximized over all possible data sets). From this, it is clear that $\mathfrak{s} \geq d$. Indeed, any set $\{x_1, \ldots, x_k\}$ shatterable by $\mathbb{C}$ is also a star set for $\mathbb{C}$, since some $h_0 \in \mathbb{C}$ classifies all k points -1, and for each x_i, some $h_i \in \mathbb{C}$ has $h_i(x_i) = +1$ while $h_i(x_j) = -1$ for every $j \neq i$ (where h_i is guaranteed to exist by shatterability of the set). On the other hand, there is no general upper bound on $\mathfrak{s}$ in terms of d, and the gap between $\mathfrak{s}$ and d can generally be infinite.

4. A similar notion previously appeared in a lower-bound argument of Dasgupta (2005), including a kind of distribution-dependent version of the "star set" idea. Indeed, we explore these connections formally in Section 7, where we additionally prove this definition is exactly equivalent to a quantity studied by Hanneke (2007a) (namely, the distribution-free version of the extended teaching dimension growth function), and has connections to several other complexity measures in the literature.

5. The maximum degree in the one-inclusion graph was recently studied in the context of teaching complexity by Fan (2012). However, using the data-induced one-inclusion graph of Haussler, Littlestone, and Warmuth (1994) (rather than the graph based on the full space $\mathcal{X}$) can substantially increase the maximum degree by omitting certain highly-informative points.

4.1 Examples

Before continuing, we briefly go through a few simple example calculations of the star number. For the class of *threshold* classifiers on $\mathbb{R}$ (i.e., $\mathbb{C} = \{x \mapsto 2\mathbb{1}_{[t,\infty)}(x) - 1 : t \in \mathbb{R}\}$), we have $\mathfrak{s} = 2$, as $\{x_1, x_2\}$ is a star set for $\mathbb{C}$ centered at $2\mathbb{1}_{[t,\infty)} - 1$ if and only if $x_1 < t \leq x_2$, and any set $\{x_1, x_2, x_3\}$ cannot be a star set for $\mathbb{C}$ centered at any given $2\mathbb{1}_{[t,\infty)} - 1$ since, of the (at least) two of these points on the same side of t, any threshold classifier disagreeing with $2\mathbb{1}_{[t,\infty)} - 1$ on the one further from t must also disagree with $2\mathbb{1}_{[t,\infty)} - 1$ on the one closer to t. In contrast, for the class of *interval* classifiers on $\mathbb{R}$ (i.e., $\mathbb{C} = \{x \mapsto 2\mathbb{1}_{[a,b]}(x) - 1 : -\infty < a \leq b < \infty\}$), we have $\mathfrak{s} = \infty$, since for *any* distinct points $x_0, x_1, \ldots, x_s \in \mathbb{R}$, $\{x_1, \ldots, x_s\}$ is a star set for $\mathbb{C}$ witnessed by $\{2\mathbb{1}_{[x_0,x_0]} - 1, 2\mathbb{1}_{[x_1,x_1]} - 1, \ldots, 2\mathbb{1}_{[x_s,x_s]} - 1\}$. It is an easy exercise to verify that we also have $\mathfrak{s} = \infty$ for the classes of *linear separators* on $\mathbb{R}^k$ ($k \geq 2$) and axis-aligned rectangles on $\mathbb{R}^k$ ($k \geq 1$), since the above construction for interval classifiers can be embedded into these spaces, with the star set lying within a lower-dimensional manifold in $\mathbb{R}^k$ (see Dasgupta, 2004, 2005; Hanneke, 2014).

As an intermediate case, where $\mathfrak{s}$ has a range of values, consider the class of *intervals of width at least* $w \in (0,1)$ (i.e., $\mathbb{C} = \{x \mapsto 2\mathbb{1}_{[a,b]}(x) - 1 : -\infty < a \leq b < \infty, b - a \geq w\}$), for the space $\mathcal{X} = [0,1]$. In this case, we can show that $\lfloor 2/w \rfloor \leq \mathfrak{s} \leq \lfloor 2/w \rfloor + 2$, as follows. We may note that letting $k = \lfloor 2/(w+\varepsilon) \rfloor + 1$ (for $\varepsilon > 0$), and taking $x_i = (w+\varepsilon)(i-1)/2$ for $1 \leq i \leq k$, we have that $\{x_1, \ldots, x_k\}$ is a star set for $\mathbb{C}$, witnessed by $\{2\mathbb{1}_{[-2w,-w]} - 1, 2\mathbb{1}_{[x_1 - w/2, x_1 + w/2]} - 1, \ldots, 2\mathbb{1}_{[x_k - w/2, x_k + w/2]} - 1\}$. Thus, taking $\varepsilon \to 0$ reveals that $\mathfrak{s} \geq \lfloor 2/w \rfloor$. On the other hand, for any $k' \in \mathbb{N}$ with $k' > 2$, and points $x_1, \ldots, x_{k'} \in [0,1]$, suppose $\{x_1, \ldots, x_{k'}\}$ is a star set for $\mathbb{C}$ witnessed by $\{h_0, h_1, \ldots, h_{k'}\}$. Without loss of generality, suppose $x_1 \leq x_2 \leq \cdots \leq x_{k'}$. First suppose h_0 classifies all of these points -1. Note that, for any $i \in \{3, \ldots, k'\}$, since the interval corresponding to h_{i-1} has width at least w and contains x_{i-1} but not x_{i-2} or x_i, we have $x_i - x_{i-1} > \max\{0, w - (x_{i-1} - x_{i-2})\}$. Thus, $1 \geq \sum_{i=2}^{k'} x_i - x_{i-1} > x_2 - x_1 + \sum_{i=3}^{k'} \max\{0, w - (x_{i-1} - x_{i-2})\} \geq (k'-2)w - \sum_{i=3}^{k'-1} x_i - x_{i-1} = (k'-2)w - (x_{k'-1} - x_2)$, so that $x_{k'-1} - x_2 > (k'-2)w - 1$. But $x_{k'-1} - x_2 \leq 1$, so that $k' < 2/w + 2$. Since k' is an integer, this implies $k' \leq \lfloor 2/w \rfloor + 2$. For the remaining case, if h_0 classifies some x_i as $+1$, then let $x_{i_0} = \min\{x_i : h_0(x_i) = +1\}$ and $x_{i_1} = \max\{x_i : h_0(x_i) = +1\}$. Note that, if $i_0 > 1$, then for any $x < x_{i_0 - 1}$, any $h \in \mathbb{C}$ with $h(x_{i_0}) = h(x) = +1 \neq h_0(x)$ must have $h(x_{i_0 - 1}) = +1 \neq h_0(x_{i_0 - 1})$, so that $\{x, x_{i_0 - 1}\} \subseteq \mathrm{DIS}(\{h, h_0\})$. Therefore, $\nexists x_i < x_{i_0 - 1}$ (since otherwise $\mathrm{DIS}(\{h_i, h_0\}) \cap \{x_1, \ldots, x_{k'}\} = \{x_i\}$ would be violated), so that $i_0 \leq 2$. Symmetric reasoning implies $i_1 \geq k' - 1$. Similarly, if $\exists x \in (x_{i_0}, x_{i_1})$, then any $h \in \mathbb{C}$ with $h(x) = -1 \neq h_0(x)$ must have either $h(x_{i_0}) = -1 \neq h_0(x_{i_0})$ or $h(x_{i_1}) = -1 \neq h_0(x_{i_1})$, so that either $\{x, x_{i_0}\} \subseteq \mathrm{DIS}(\{h, h_0\})$ or $\{x, x_{i_1}\} \subseteq \mathrm{DIS}(\{h, h_0\})$. Therefore, $\nexists x_i \in (x_{i_0}, x_{i_1})$ (since again, $\mathrm{DIS}(\{h_i, h_0\}) \cap \{x_1, \ldots, x_{k'}\} = \{x_i\}$ would be violated), so that $i_1 \in \{i_0, i_0 + 1\}$. Combined, these facts imply $k' \leq i_1 + 1 \leq i_0 + 2 \leq 4 \leq \lfloor 2/w \rfloor + 2$. Altogether, we have $\mathfrak{s} \leq \lfloor 2/w \rfloor + 2$.

5. Main Results

We are now ready to state the main results of this article: upper and lower bounds on the minimax label complexities under the above noise models. For the sake of making the theorem statements more concise, we abstract the dependence on logarithmic factors in

several of the upper bounds into a simple "polylog(x)" factor, meaning a value $\lesssim \mathrm{Log}^k(x)$, for some $k \in [1,\infty)$ (in fact, all of these results hold with values of $k \leq 4$); the reader is referred to the proofs for a description of the actual logarithmic factors this polylog function represents, along with tighter expressions of the upper bounds. The formal proofs of all of these results are included in Appendix B.

Theorem 3 *For any $\varepsilon \in (0,1/9)$, $\delta \in (0,1/3)$,*

$$\max\left\{\min\left\{\mathfrak{s}, \frac{1}{\varepsilon}\right\}, d, \mathrm{Log}\left(\min\left\{\frac{1}{\varepsilon}, |\mathbb{C}|\right\}\right)\right\} \lesssim \Lambda_{\mathrm{RE}}(\varepsilon,\delta) \lesssim \min\left\{\mathfrak{s}, \frac{d}{\varepsilon}, \frac{\mathfrak{s}d}{\mathrm{Log}(\mathfrak{s})}\right\} \mathrm{Log}\left(\frac{1}{\varepsilon}\right).$$

Theorem 4 *For any $\beta \in [0,1/2)$, $\varepsilon \in (0,(1-2\beta)/24)$, $\delta \in (0,1/24]$,*

$$\frac{1}{(1-2\beta)^2}\max\left\{\min\left\{\mathfrak{s}, \frac{1-2\beta}{\varepsilon}\right\}\beta\mathrm{Log}\left(\frac{1}{\delta}\right), d\right\} \lesssim \Lambda_{\mathrm{BN}(\beta)}(\varepsilon,\delta) \lesssim \frac{1}{(1-2\beta)^2}\min\left\{\mathfrak{s}, \frac{(1-2\beta)d}{\varepsilon}\right\}\mathrm{polylog}\left(\frac{d}{\varepsilon\delta}\right).$$

Theorem 5 *For any $a \in [4,\infty)$, $\alpha \in (0,1)$, $\varepsilon \in (0,1/(24a^{1/\alpha}))$, and $\delta \in (0,1/24]$, if $0 < \alpha \leq 1/2$,*

$$a^2\left(\frac{1}{\varepsilon}\right)^{2-2\alpha}\left(d + \mathrm{Log}\left(\frac{1}{\delta}\right)\right) \lesssim \Lambda_{\mathrm{TN}(a,\alpha)}(\varepsilon,\delta) \lesssim a^2\left(\frac{1}{\varepsilon}\right)^{2-2\alpha} d\cdot\mathrm{polylog}\left(\frac{d}{\varepsilon\delta}\right)$$

and if $1/2 < \alpha < 1$,

$$a^2\left(\frac{1}{\varepsilon}\right)^{2-2\alpha}\max\left\{\min\left\{\mathfrak{s}, \frac{1}{a^{1/\alpha}\varepsilon}\right\}^{2\alpha-1}\mathrm{Log}\left(\frac{1}{\delta}\right), d\right\} \lesssim \Lambda_{\mathrm{TN}(a,\alpha)}(\varepsilon,\delta) \lesssim a^2\left(\frac{1}{\varepsilon}\right)^{2-2\alpha}\min\left\{\frac{\mathfrak{s}}{d}, \frac{1}{a^{1/\alpha}\varepsilon}\right\}^{2\alpha-1} d\cdot\mathrm{polylog}\left(\frac{d}{\varepsilon\delta}\right).$$

Theorem 6 *For any $a \in [4,\infty)$, $\alpha \in (0,1)$, $\varepsilon \in (0,1/(24a^{1/\alpha}))$, and $\delta \in (0,1/24]$, if $0 \leq \alpha \leq 1/2$,*

$$a^2\left(\frac{1}{\varepsilon}\right)^{2-2\alpha}\left(d + \mathrm{Log}\left(\frac{1}{\delta}\right)\right) \lesssim \Lambda_{\mathrm{BC}(a,\alpha)}(\varepsilon,\delta) \lesssim a^2\left(\frac{1}{\varepsilon}\right)^{2-2\alpha}\min\left\{\mathfrak{s}, \frac{1}{a\varepsilon^\alpha}\right\} d\cdot\mathrm{polylog}\left(\frac{1}{\varepsilon\delta}\right),$$

and if $1/2 < \alpha \leq 1$,

$$a^2\left(\frac{1}{\varepsilon}\right)^{2-2\alpha}\max\left\{\min\left\{\mathfrak{s}, \frac{1}{a^{1/\alpha}\varepsilon}\right\}^{2\alpha-1}\mathrm{Log}\left(\frac{1}{\delta}\right), d\right\} \lesssim \Lambda_{\mathrm{BC}(a,\alpha)}(\varepsilon,\delta) \lesssim a^2\left(\frac{1}{\varepsilon}\right)^{2-2\alpha}\min\left\{\mathfrak{s}, \frac{1}{a\varepsilon^\alpha}\right\} d\cdot\mathrm{polylog}\left(\frac{1}{\varepsilon\delta}\right).$$

Theorem 7 *For any $\nu \in [0,1/2)$, $\varepsilon \in (0,(1-2\nu)/24)$, and $\delta \in (0,1/24]$,*

$$\frac{\nu^2}{\varepsilon^2}\left(d + \mathrm{Log}\left(\frac{1}{\delta}\right)\right) + \min\left\{\mathfrak{s}, \frac{1}{\varepsilon}\right\} \lesssim \Lambda_{\mathrm{BE}(\nu)}(\varepsilon,\delta) \lesssim \left(\frac{\nu^2}{\varepsilon^2}d + \min\left\{\mathfrak{s}, \frac{d}{\varepsilon}\right\}\right)\mathrm{polylog}\left(\frac{d}{\varepsilon\delta}\right).$$

Theorem 8 *For any $\nu \in [0, 1/2)$, $\varepsilon \in (0, (1-2\nu)/24)$, and $\delta \in (0, 1/24]$,*

$$\frac{\nu^2}{\varepsilon^2}\left(d + \operatorname{Log}\left(\frac{1}{\delta}\right)\right) + \min\left\{\mathfrak{s}, \frac{1}{\varepsilon}\right\} \lesssim \Lambda_{\mathrm{AG}(\nu)}(\varepsilon, \delta) \lesssim \min\left\{\mathfrak{s}, \frac{1}{\nu+\varepsilon}\right\}\left(\frac{\nu^2}{\varepsilon^2}+1\right) d \cdot \operatorname{polylog}\left(\frac{1}{\varepsilon\delta}\right).$$

5.1 Remarks on the Main Results

We sketch the main innovations underlying the active learning algorithms achieving these upper bounds in Section 5.2 below. Sections 6 and 7 then provide detailed and thorough comparisons of each of these results to those in the prior literature on passive and active learning. For now, we mention a few noteworthy observations and comments regarding these theorems.

5.1.1 Comparison to the Previous Best Known Results

Aside from Theorems 6 and 8, each of the above results offers some kind of refinement over the previous best known results on the label complexity of active learning. Some of these refinements are relatively mild, such as those for the realizable case and bounded noise. However, our refinements under Tsybakov noise and benign noise are far more significant. In particular, perhaps the most surprising and interesting of the above results are the upper bounds in Theorem 5, which can be considered the primary contribution of this work.

As discussed above, the prior literature on noise-robust active learning is largely rooted in the intuitions and techniques developed for the realizable case. As indicated by Theorem 3, there is a wide spread of label complexities for active learning problems in the realizable case, depending on the structure of the hypothesis class. In particular, when $\mathfrak{s} < \infty$, we have $O(\operatorname{Log}(1/\varepsilon))$ label complexity in the realizable case, representing a nearly-exponential improvement over passive learning, which has $\tilde{\Theta}(1/\varepsilon)$ dependence on ε. On the other hand, when $\mathfrak{s} = \infty$, we have $\Omega(1/\varepsilon)$ minimax label complexity for active learning, which is the same dependence on ε as known for passive learning (see Section 6). Thus, for active learning in the realizable case, some hypothesis classes are "easy" (such as threshold classifiers), offering strong improvements over passive learning, while others are "hard" (such as interval classifiers), offering almost no improvements over passive.

With the realizable case as inspiration, the results in the prior literature on general noise-robust active learning have all continued to reflect these distinctions, and the label complexity bounds in those works continue to exhibit this wide spread. In the case of Tsybakov noise, the best general results in the prior literature (from Hanneke and Yang, 2012; Hanneke, 2014) correspond to an upper bound of roughly $a^2\left(\frac{1}{\varepsilon}\right)^{2-2\alpha} \min\left\{\mathfrak{s}, \frac{1}{a\varepsilon^\alpha}\right\} d \cdot \operatorname{polylog}\left(\frac{1}{\varepsilon\delta}\right)$ (after converting those complexity measures into the star number via the results in Section 7 below). When $\mathfrak{s} < \infty$, this has dependence $\tilde{\Theta}(\varepsilon^{2\alpha-2})$ on ε, which reflects a strong improvement over the $\tilde{\Theta}(\varepsilon^{\alpha-2})$ minimax sample complexity of passive learning for this problem (see Section 6). On the other hand, when $\mathfrak{s} = \infty$, this bound is $\tilde{\Theta}(\varepsilon^{\alpha-2})$, so that as in the realizable case, the bound is no better than that of passive learning for these hypothesis classes. Thus, the prior results in the literature continue the trend observed in the realizable case, in which the "easy" hypothesis classes admit strong improvements over

passive learning, while the "hard" hypothesis classes have a bound that is no better than the sample complexity of passive learning.

With this as background, it comes as quite a surprise that the upper bounds in Theorem 5 are *always* smaller than the corresponding minimax sample complexities of passive learning, in terms of their asymptotic dependence on ε for $0 < \alpha < 1$. Specifically, these upper bounds reveal a label complexity $\tilde{O}(\varepsilon^{2\alpha-2})$ when $\mathfrak{s} < \infty$, and $\tilde{O}(\varepsilon^{2\alpha-2} \vee (1/\varepsilon))$ when $\mathfrak{s} = \infty$. Comparing to the $\tilde{\Theta}(\varepsilon^{\alpha-2})$ minimax sample complexity of passive learning, the improvement for active learning is by a factor of $\tilde{\Theta}(\varepsilon^{-\alpha})$ when $\mathfrak{s} < \infty$, and by a factor of $\tilde{\Theta}(\varepsilon^{-\min\{\alpha,1-\alpha\}})$ when $\mathfrak{s} = \infty$. As a further surprise, when $0 < \alpha \leq 1/2$ (the high-noise regime), we see that the distinctions between active learning problems of a given VC dimension essentially *vanish* (up to logarithmic factors), so that the familiar spread of label complexities from the realizable case is no longer present. Indeed, in this latter case, *all* hypothesis classes with finite VC dimension exhibit the strong improvements over passive learning, previously only known to hold for the "easy" hypothesis classes (such as threshold classifiers): that is, $\tilde{O}(\varepsilon^{2\alpha-2})$ label complexity.

Further examining these upper bounds, we see that the spread of label complexities between "easy" and "hard" hypothesis classes increasingly re-emerges as α approaches 1, beginning with $\alpha = 1/2$. This transition point is quite sensible, since this is precisely the point at which the label complexity has dependence on ε of $\tilde{\Theta}(1/\varepsilon)$, which is roughly the same as the minimax label complexity of the "hard" hypothesis classes in the realizable case, which is, after all, included in $\mathrm{TN}(a, \alpha)$. Thus, as α increases above $1/2$, the "easy" hypothesis classes (with $\mathfrak{s} < \infty$) exhibit stronger improvements over passive learning, while the "hard" hypothesis classes (with $\mathfrak{s} = \infty$) continue to exhibit precisely this $\tilde{\Theta}\left(\frac{1}{\varepsilon}\right)$ behavior. In either case, the label complexity exhibits an improvement in dependence on ε compared to passive learning for the same α value. But since the label complexity of passive learning decreases to $\tilde{\Theta}\left(\frac{1}{\varepsilon}\right)$ as $\alpha \to 1$, we naturally have that for the "hard" hypothesis classes, the gap between the passive and active label complexities shrinks as α approaches 1. In contrast, the "easy" hypothesis classes exhibit a gap between passive and active label complexities that becomes more pronounced as α approaches 1 (with a near-exponential improvement over passive learning exhibited in the limiting case, corresponding to bounded noise).

This same pattern is present, though to a lesser extent, for benign noise. In this case, the best general results in the prior literature (from Dasgupta, Hsu, and Monteleoni, 2007; Hanneke, 2007a, 2014) correspond to an upper bound of roughly $\min\left\{\mathfrak{s}, \frac{1}{\nu+\varepsilon}\right\}\left(\frac{\nu^2}{\varepsilon^2}+1\right) d \cdot \mathrm{polylog}\left(\frac{1}{\varepsilon\delta}\right)$ (again, after converting those complexity measures into the star number via the results in Section 7 below). When $\mathfrak{s} < \infty$, the dependence on ν and ε is roughly $\tilde{\Theta}\left(\frac{\nu^2}{\varepsilon^2}\right)$ (aside from logarithmic factors and constants, and for $\nu > \varepsilon$). However, when $\mathfrak{s} = \infty$, this dependence becomes roughly $\tilde{\Theta}\left(\frac{\nu}{\varepsilon^2}\right)$, which is the same as in the minimax sample complexity of passive learning (see Section 6). Thus, for these results in the prior literature, we again see that the "easy" hypothesis classes have a bound reflecting improvements over passive learning, while the bound for the "hard" hypothesis classes fail to reflect any improvements over passive learning at all.

In contrast, consider the upper bound in Theorem 7. In this case, when $\nu \geq \sqrt{\varepsilon}$ (again, the high-noise regime), for *all* hypothesis classes with finite VC dimension, the dependence on ν and ε is roughly $\tilde{\Theta}\left(\frac{\nu^2}{\varepsilon^2}\right)$. Again, this makes almost no distinction between "easy"

hypothesis classes (with $\mathfrak{s} < \infty$) and "hard" hypothesis classes (with $\mathfrak{s} = \infty$), and instead always exhibits the strongest possible improvements (up to logarithmic factors), previously only known to hold for the "easy" classes (such as threshold classifiers): namely, reduction in label complexity by roughly a factor of $1/\nu$ compared to passive learning. The improvements in this case are typically milder than we found in Theorem 5, but noteworthy nonetheless. Again, as ν decreases below $\sqrt{\varepsilon}$, the distinction between "easy" and "hard" hypothesis classes begins to re-emerge, with the harder classes maintaining a $\tilde{\Theta}\left(\frac{1}{\varepsilon}\right)$ dependence (roughly equivalent to the realizable-case label complexity for these classes), while the easier classes continue to exhibit the $\tilde{\Theta}\left(\frac{\nu^2}{\varepsilon^2}\right)$ behavior, approaching $O\left(\text{polylog}\left(\frac{1}{\varepsilon}\right)\right)$ as ν shrinks.

5.1.2 The Dependence on δ

One remarkable fact about $\Lambda_{\text{RE}}(\varepsilon, \delta)$ is that there is *no* significant dependence on δ in the optimal label complexity for the given range of δ.[6] Note that this is not the case in noisy settings, where the lower bounds have an explicit dependence on δ. In the proofs, this dependence on δ is introduced via randomness of the labels. However, as argued by Kääriäinen (2006), a dependence on δ is sometimes still required in $\Lambda_{\mathbb{D}}(\varepsilon, \delta)$, even if we restrict $\mathbb{D}$ to those $\mathcal{P}_{XY} \in \text{AG}(\nu)$ inducing *deterministic* labels: that is, $\eta(x; \mathcal{P}_{XY}) \in \{0, 1\}$ for all x.

5.1.3 Spanning the Gaps

All of these results have gaps between the lower and upper bounds. It is interesting to note that one can construct examples of hypothesis classes spanning these gaps, for Theorems 3, 4, 5, and 7 (up to logarithmic factors). For instance, for sufficiently large d and $\mathfrak{s}$ and sufficiently small ε and δ, these upper bounds are tight (up to logarithmic factors) in the case where $\mathbb{C} = \{x \mapsto 2\mathbb{1}_S(x) - 1 : S \subseteq \{1, \ldots, \mathfrak{s}\}, |S| \leq d\}$, for $\mathcal{X} = \mathbb{N}$ (taking inspiration from a suggested modification by Hanneke, 2014, of the proof of a related result of Raginsky and Rakhlin, 2011). Likewise, these lower bounds are tight (up to logarithmic factors) in the case that $\mathcal{X} = \mathbb{N}$ and $\mathbb{C} = \{x \mapsto 2\mathbb{1}_S(x) - 1 : S \in 2^{\{1,\ldots,d\}} \cup \{\{i\} : d+1 \leq i \leq \mathfrak{s}\}\}$.[7] Thus, these upper and lower bounds cannot be significantly refined (without loss of generality) without introducing additional complexity measures to distinguish these cases. For completeness, we include proofs of these claims in Appendix D. It immediately follows from this (and monotonicity of the respective noise models in $\mathbb{C}$) that the upper and lower bounds in Theorems 3, 4, 5, and 7 are each sometimes tight in the case $\mathfrak{s} = \infty$, as limiting cases of the above constructions: that is, the upper bounds are tight (up to logarithmic factors) for $\mathbb{C} = \{x \mapsto 2\mathbb{1}_S(x) - 1 : S \subseteq \mathbb{N}, |S| \leq d\}$, and the lower bounds are tight (up to logarithmic factors) for $\mathbb{C} = \{x \mapsto 2\mathbb{1}_S(x) - 1 : S \in 2^{\{1,\ldots,d\}} \cup \{\{i\} : d+1 \leq i < \infty\}\}$. It is interesting to note that the above space $\mathbb{C}$ for which the upper bounds are tight can be embedded in a variety of hypothesis classes in common use in machine learning (while maintaining VC

6. We should expect a more significant dependence on δ near 1, since one case easily prove that $\Lambda_{\text{RE}}(\varepsilon, \delta) \to 0$ as $\delta \to 1$.
7. Technically, for Theorems 4 and 7, we require slightly stronger versions of the lower bound to establish tightness for β or ν near 0: namely, adding the lower bound from Theorem 3 to these lower bounds. The validity of this stronger lower bound follows immediately from the facts that $\text{RE} \subseteq \text{BN}(\beta)$ and $\text{RE} \subseteq \text{BE}(\nu)$.

dimension $\lesssim d$ and star number $\lesssim \mathfrak{s}$): for instance, in the case of $\mathfrak{s} = \infty$, this is true of linear separators in $\mathbb{R}^{3d}$ and axis-aligned rectangles in $\mathbb{R}^{2d}$. It follows that the upper bounds in these theorems are tight (up to logarithmic factors) for each of these hypothesis classes.

5.1.4 Separation of $\mathrm{TN}(a,\alpha)$ and $\mathrm{BC}(a,\alpha)$

Another interesting implication of these results is a separation between the noise models $\mathrm{TN}(a,\alpha)$ and $\mathrm{BC}(a,\alpha)$ not previously noted in the literature. Specifically, if we consider any class $\mathbb{C}$ comprised of only the $\mathfrak{s}+1$ classifiers in Definition 2, then one can show[8] that (for $\mathfrak{s} \geq 3$), for any $\alpha \in (0,1]$, $a \in [4,\infty)$, $\varepsilon \in (0, 1/(4a^{1/\alpha}))$, and $\delta \in (0, 1/16]$,

$$\Lambda_{\mathrm{BC}(a,\alpha)}(\varepsilon,\delta) \gtrsim a^2 \left(\frac{1}{\varepsilon}\right)^{2-2\alpha} \min\left\{\mathfrak{s}, \frac{1}{a\varepsilon^\alpha}\right\} \mathrm{Log}\left(\frac{1}{\delta}\right).$$

In particular, when $\mathfrak{s} > \frac{1}{a\varepsilon^\alpha}$, we have $\Lambda_{\mathrm{BC}(a,\alpha)}(\varepsilon,\delta) \gtrsim a\varepsilon^{\alpha-2}\mathrm{Log}(1/\delta)$, which is larger than the upper bound on $\Lambda_{\mathrm{TN}(a,\alpha)}(\varepsilon,\delta)$. Furthermore, when $\mathfrak{s} = \infty$, this lower bound has asymptotic dependence on ε that is $\Omega(\varepsilon^{\alpha-2})$, which is the same dependence found in the sample complexity of passive learning, up to a logarithmic factor (see Section 6 below). Comparing this to the upper bounds in Theorem 5, which exhibit asymptotic dependence on ε as $\Lambda_{\mathrm{TN}(a,\alpha)}(\varepsilon,\delta) = \tilde{O}(\varepsilon^{\min\{2\alpha-1,0\}-1})$ when $\mathfrak{s} = \infty$, we see that for this class, any $\alpha \in (0,1)$ has $\Lambda_{\mathrm{TN}(a,\alpha)}(\varepsilon,\delta) \ll \Lambda_{\mathrm{BC}(a,\alpha)}(\varepsilon,\delta)$. One reason this separation is interesting is that most of the existing literature on active learning under $\mathrm{TN}(a,\alpha)$ makes use of the noise condition via the fact that it implies $\mathcal{P}(x : h(x) \neq f^\star_{\mathcal{P}_{XY}}(x)) \leq a(\mathrm{er}_{\mathcal{P}_{XY}}(h) - \mathrm{er}_{\mathcal{P}_{XY}}(f^\star_{\mathcal{P}_{XY}}))^\alpha$ for all $h \in \mathbb{C}$: that is, $\mathrm{TN}(a,\alpha) \subseteq \mathrm{BC}(a,\alpha)$. This separation indicates that, to achieve the optimal performance under $\mathrm{TN}(a,\alpha)$, one needs to consider more-specific properties of this noise model, beyond those satisfied by $\mathrm{BC}(a,\alpha)$. Another reason this separation is quite interesting is that it contrasts with the known results for *passive* learning, where (as we discuss in Section 6 below) the sample complexities under these two noise models are *equivalent* (up to an unresolved logarithmic factor).

5.1.5 Gaps in Theorems 6 and 8, and Related Open Problems

We conjecture that the dependence on d and $\mathfrak{s}$ in the upper bounds of Theorem 6 can be refined in general (where presently it is linear in $\mathfrak{s}d$). More specifically, we conjecture that the upper bound can be improved to

$$\Lambda_{\mathrm{BC}(a,\alpha)}(\varepsilon,\delta) \lesssim a^2 \left(\frac{1}{\varepsilon}\right)^{2-2\alpha} \min\left\{\mathfrak{s}, \frac{d}{a\varepsilon^\alpha}\right\} \mathrm{polylog}\left(\frac{1}{\varepsilon\delta}\right),$$

though it is unclear at this time as to how this might be achieved. The above example (separating $\mathrm{BC}(a,\alpha)$ from $\mathrm{TN}(a,\alpha)$) indicates that we generally cannot hope to reduce the upper bound on the label complexity for $\mathrm{BC}(a,\alpha)$ much beyond this.

As for whether the form of the upper bound on $\Lambda_{\mathrm{AG}(\nu)}(\varepsilon,\delta)$ in Theorem 8 can generally be improved to match the form of the upper bound for $\Lambda_{\mathrm{BE}(\nu)}(\varepsilon,\delta)$, this remains a fascinating open question. We conjecture that at least the dependence on d and $\mathfrak{s}$ can be improved to some extent (where presently it is linear in $d\mathfrak{s}$).

8. Specifically, this follows by taking $\zeta = \frac{a}{2}(4\varepsilon)^\alpha$, $\beta = \frac{1}{2} - \frac{2}{a4^\alpha}\varepsilon^{1-\alpha}$, and $k = \min\{\mathfrak{s}-1, \lfloor 1/\zeta \rfloor\}$ in Lemma 26 of Appendix A.2, and noting that the resulting set of distributions $\mathrm{RR}(k,\zeta,\beta)$ is contained in $\mathrm{BC}(a,\alpha)$ for this $\mathbb{C}$.

5.1.6 Minutiae

We note that the restrictions to the ranges of ε and δ in the above results are required only for the lower bounds (aside from $\delta \in (0,1]$, $\varepsilon > 0$), as are the restrictions to the ranges of the parameters a, α, and ν, aside from the constraints in the definitions in Section 3; the upper bounds are proven without any such restrictions in Appendix B. Also, several of the upper bounds above (e.g., Theorems 5 and 7) are slightly looser (by logarithmic factors) than those actually proven in Appendix B, which are typically stated in a different form (e.g., with factors of $d\mathrm{Log}\left(\frac{1}{\varepsilon}\right) + \mathrm{Log}\left(\frac{1}{\delta}\right)$, rather than simply $d \cdot \mathrm{polylog}\left(\frac{1}{\varepsilon\delta}\right)$). We state the weaker results here purely to simplify the theorem statements, referring the interested reader to the proofs for the refined versions. However, aside from Theorem 3, we believe it is possible to further optimize the logarithmic factors in all of these upper bounds.

We additionally note that we can also obtain results by the subset relations between the noise models. For instance, since $\mathrm{RE} \subseteq \mathrm{BN}(\beta) \subseteq \mathrm{BE}(\beta) \subseteq \mathrm{AG}(\beta)$, in the case β is close to 0 we can increase the lower bounds in Theorems 4, 7, and 8 based on the lower bound in Theorem 3: that is, for $\nu \geq \beta \geq 0$,

$$\Lambda_{\mathrm{AG}(\nu)}(\varepsilon,\delta) \geq \Lambda_{\mathrm{BE}(\nu)}(\varepsilon,\delta) \geq \Lambda_{\mathrm{BN}(\beta)}(\varepsilon,\delta) \geq \Lambda_{\mathrm{RE}}(\varepsilon,\delta) \gtrsim \max\left\{\min\left\{\mathfrak{s}, \frac{1}{\varepsilon}\right\}, d\right\}.$$

Similarly, since RE is contained in all of the noise models studied here, $\mathrm{Log}\left(\min\left\{\frac{1}{\varepsilon}, |\mathbb{C}|\right\}\right)$ can also be included as a lower bound in each of these results. Likewise, in the cases that a is very large or α is very close to 0, we can get a more informative upper bound in Theorem 5 via Theorem 7, since $\mathrm{TN}(a,\alpha) \subseteq \mathrm{BE}(1/2)$. For simplicity, in most of the above theorems, we have not explicitly included the various compositions of the above results that can be obtained in this way (with only a few exceptions).

5.2 The Strategy behind Theorems 5 and 7

The upper bounds in Theorems 5 and 7 represent the main results of this work, and along with the upper bound in Theorem 4, are based on a general argument with essentially three main components. The first component is a more-sophisticated variant of a basic approach introduced to the active learning literature by Kääriäinen (2006): namely, reduction to the realizable case via repeatedly querying for the label at a point in $\mathcal{X}$ until its Bayes optimal classification can be determined (based on a sequential probability ratio test, as studied by Wald, 1945, 1947). Of course, in the present model of active learning, repeatedly requesting a label Y_i yields no new information beyond requesting Y_i once, since we are not able to resample from the distribution of Y_i given X_i (as Kääriäinen, 2006, does). To resolve this, we argue that it is possible to partition the space $\mathcal{X}$ into cells, in a way such that $f^\star_{\mathcal{P}_{XY}}$ is nearly constant in the vast majority of cells (without direct knowledge of $f^\star_{\mathcal{P}_{XY}}$ or $\mathcal{P}$); this is essentially a data-dependent approximation to the recently-discovered finite approximability property of VC classes (Adams and Nobel, 2012). Given this partition, for a given point X_i, we can find many other points X_j in the same cell of the partition as X_i, and request labels for these points until we can determine what the majority label for the cell is. We show that, with high probability, this value will equal $f^\star_{\mathcal{P}_{XY}}(X_i)$, so that we can effectively use these majority labels in an active learning algorithm for the realizable case.

However, we note that in the case of $\mathrm{TN}(a,\alpha)$, if we simply apply this repeated querying strategy to random $\mathcal{P}$-distributed samples, the resulting label complexity would be too large, and we would sometimes expect to exhaust most of the queries determining the optimal labels in very *noisy* regions (i.e., in cells of the partition where $\eta(\cdot;\mathcal{P}_{XY})$ is close to $1/2$ on average). This is because Tsybakov's condition allows that such regions can have non-negligible probability, and the number of samples required to determine the majority value of a ± 1 random variable becomes unbounded as its mean approaches zero. However, we can note that it is also less important for the final classifier $\hat{h}$ to agree with $f^\star_{\mathcal{P}_{XY}}$ on these high-noise points than it is for low-noise points, since classifying them opposite from $f^\star_{\mathcal{P}_{XY}}$ has less impact on the excess error rate $\mathrm{er}_{\mathcal{P}_{XY}}(\hat{h}) - \mathrm{er}_{\mathcal{P}_{XY}}(f^\star_{\mathcal{P}_{XY}})$. Therefore, as the second main component of our active learning strategy, we take a tiered approach to learning, effectively shifting the distribution $\mathcal{P}$ to favor points in cells with average $\eta(\cdot;\mathcal{P}_{XY})$ value further from $1/2$. We achieve this by discarding a point X_i if the number of queries exhausted toward determining the majority label in its cell of the partition becomes excessively large, and we gradually decrease this threshold as the data set grows, so that the points making it through this filter have progressively less and less noisy labels. By choosing $\hat{h}$ to agree with the inferred $f^\star_{\mathcal{P}_{XY}}$ classification of every point passing this filter, and combining this with the standard analysis of learning in the realizable case (Vapnik, 1982, 1998; Blumer, Ehrenfeucht, Haussler, and Warmuth, 1989), this allows us to provide a bound on the fraction of points in $\mathcal{X}$ at a given level of noisiness (i.e., $|\eta(\cdot;\mathcal{P}_{XY}) - 1/2|$) on which the produced classifier $\hat{h}$ disagrees with $f^\star_{\mathcal{P}_{XY}}$, such that this bound decreases as the noisiness decreases (i.e., as $|\eta(\cdot;\mathcal{P}_{XY}) - 1/2|$ increases). Furthermore, by discarding many of the points in high-noise regions without exhausting too many label requests trying to determine their $f^\star_{\mathcal{P}_{XY}}$ classifications, we are able to reduce the total number of label requests needed to obtain ε excess error rate.

Already these two components comprise the essential strategy that achieves these upper bounds in the case of $\mathfrak{s} = \infty$. However, to obtain the stated dependence on $\mathfrak{s}$ in these bounds when $\mathfrak{s} < \infty$, we need to introduce a third component: namely, using the inferred values of $f^\star_{\mathcal{P}_{XY}}(X_i)$ in the context of an active learning algorithm for the realizable case. For this, we specifically use the disagreement-based strategy of Cohn, Atlas, and Ladner (1994) (known as CAL), which processes the unlabeled data in sequence, and requests to observe the classification $f^\star_{\mathcal{P}_{XY}}(X_i)$ if and only if X_i is in the region of disagreement of the set of classifiers in $\mathbb{C}$ consistent with all previously-observed $f^\star_{\mathcal{P}_{XY}}(X_j)$ values. Using a modification of a recent analysis of this algorithm by Wiener, Hanneke, and El-Yaniv (2015) (applied to each tier of label-noise separately), combined with the results below (in Section 7.3) relating the complexity measure used in that analysis to the star number, we obtain the dependence on $\mathfrak{s}$ stated in the above results.

6. Comparison to Passive Learning

The natural baseline for comparison in active learning is the *passive learning* protocol, in which the labeled data are i.i.d. samples with common distribution $\mathcal{P}_{XY}$: that is, the input to the passive learning algorithm is $(X_1,Y_1),\ldots,(X_n,Y_n)$. In this context, the minimax sample complexity of passive learning, denoted $\mathcal{M}_{\mathbb{D}}(\varepsilon,\delta)$, is defined as the smallest $n \in \mathbb{N} \cup \{0\}$ for which there exists a passive learning rule mapping $(X_1,Y_1),\ldots,(X_n,Y_n)$ to

a classifier $\hat{h} : \mathcal{X} \to \mathcal{Y}$ such that, for any $\mathcal{P}_{XY} \in \mathbb{D}$, with probability at least $1 - \delta$, $\mathrm{er}_{\mathcal{P}_{XY}}(\hat{h}) - \inf_{h \in \mathbb{C}} \mathrm{er}_{\mathcal{P}_{XY}}(h) \leq \varepsilon$.

Clearly $\Lambda_{\mathbb{D}}(\varepsilon, \delta) \leq \mathcal{M}_{\mathbb{D}}(\varepsilon, \delta)$ for any $\mathbb{D}$, since for every passive learning algorithm $\mathcal{A}$, there is an active learning algorithm that requests $Y_1, \ldots, Y_n$ and then runs $\mathcal{A}$ with $(X_1, Y_1), \ldots, (X_n, Y_n)$ to determine the returned classifier. One of the main interests in the theory of active learning is determining the size of the gap between these two complexities, for various sets $\mathbb{D}$. For the purpose of this comparison, we now review several results known to hold for $\mathcal{M}_{\mathbb{D}}(\varepsilon, \delta)$, for various sets $\mathbb{D}$. Specifically, the following bounds are known to hold for any choice of hypothesis class $\mathbb{C}$, and for β, a, α, ν, ε, and δ as in the respective theorems from Section 5 (Vapnik and Chervonenkis, 1971; Vapnik, 1982, 1998; Blumer, Ehrenfeucht, Haussler, and Warmuth, 1989; Ehrenfeucht, Haussler, Kearns, and Valiant, 1989; Haussler, Littlestone, and Warmuth, 1994; Massart and Nédélec, 2006; Hanneke, 2014).

- $\frac{1}{\varepsilon}\left(d + \mathrm{Log}\left(\frac{1}{\delta}\right)\right) \lesssim \mathcal{M}_{\mathrm{RE}}(\varepsilon, \delta) \lesssim \frac{1}{\varepsilon}\left(d\mathrm{Log}\left(\frac{1}{\max\{\varepsilon, \delta\}}\right) + \mathrm{Log}\left(\frac{1}{\delta}\right)\right)$.
- $\frac{1}{(1-2\beta)\varepsilon}\left(d + \mathrm{Log}\left(\frac{1}{\delta}\right)\right) \lesssim \mathcal{M}_{\mathrm{BN}(\beta)}(\varepsilon, \delta) \lesssim \frac{1}{(1-2\beta)\varepsilon}\left(d\mathrm{Log}\left(\frac{1-2\beta}{\varepsilon}\right) + \mathrm{Log}\left(\frac{1}{\delta}\right)\right)$.
- $\frac{a}{\varepsilon^{2-\alpha}}\left(d + \mathrm{Log}\left(\frac{1}{\delta}\right)\right) \lesssim \mathcal{M}_{\mathrm{TN}(a,\alpha)}(\varepsilon, \delta) \leq \mathcal{M}_{\mathrm{BC}(a,\alpha)} \lesssim \frac{a}{\varepsilon^{2-\alpha}}\left(d\mathrm{Log}\left(\frac{1}{a\varepsilon^{\alpha}}\right) + \mathrm{Log}\left(\frac{1}{\delta}\right)\right)$.
- $\frac{\nu+\varepsilon}{\varepsilon^2}\left(d + \mathrm{Log}\left(\frac{1}{\delta}\right)\right) \lesssim \mathcal{M}_{\mathrm{BE}(\nu)}(\varepsilon, \delta) \leq \mathcal{M}_{\mathrm{AG}(\nu)}(\varepsilon, \delta) \lesssim \frac{\nu+\varepsilon}{\varepsilon^2}\left(d\mathrm{Log}\left(\frac{1}{\nu+\varepsilon}\right) + \mathrm{Log}\left(\frac{1}{\delta}\right)\right)$.

Let us compare these to the results for active learning in Section 5 on a case-by-case basis. In the realizable case, we observe clear improvements of active learning over passive learning in the case $\mathfrak{s} \ll \frac{d}{\varepsilon}$ (aside from logarithmic factors). In particular, based on the upper and lower bounds for both passive and active learning, we may conclude that $\mathfrak{s} < \infty$ is necessary and sufficient for the asymptotic dependence on ε to satisfy $\Lambda_{\mathrm{RE}}(\varepsilon, \cdot) = o(\mathcal{M}_{\mathrm{RE}}(\varepsilon, \cdot))$; specifically, when $\mathfrak{s} < \infty$, $\Lambda_{\mathrm{RE}}(\varepsilon, \cdot) = O(\mathrm{Log}(\mathcal{M}_{\mathrm{RE}}(\varepsilon, \cdot)))$, and when $\mathfrak{s} = \infty$, $\Lambda_{\mathrm{RE}}(\varepsilon, \cdot) = \Theta(\mathcal{M}_{\mathrm{RE}}(\varepsilon, \cdot))$. For bounded noise, we have a similar asymptotic behavior. When $\mathfrak{s} < \infty$, again $\Lambda_{\mathrm{BN}(\beta)}(\varepsilon, \cdot) = O(\mathrm{polylog}(\mathcal{M}_{\mathrm{BN}(\beta)}(\varepsilon, \cdot)))$, and when $\mathfrak{s} = \infty$, $\Lambda_{\mathrm{BN}(\beta)}(\varepsilon, \cdot) = \tilde{\Theta}(\mathcal{M}_{\mathrm{BN}(\beta)}(\varepsilon, \cdot))$. In terms of the constants, to obtain improvements over passive learning (aside from the effects of logarithmic factors), it suffices to have $\mathfrak{s} \ll \frac{(1-2\beta)d}{\varepsilon}$, which is somewhat smaller (depending on β) than was sufficient in the realizable case.

Under Tsybakov's noise condition, every $\alpha \in (0, 1/2]$ shows an improvement in the upper bounds for active learning over the lower bound for passive learning by a factor of roughly $\frac{1}{a\varepsilon^{\alpha}}$ (aside from logarithmic factors). On the other hand, when $\alpha \in (1/2, 1)$, if $\mathfrak{s} < \frac{d}{a^{1/\alpha}\varepsilon}$, the improvement of active upper bounds over the passive lower bound is by a factor of roughly $\frac{1}{a\varepsilon^{\alpha}}\left(\frac{d}{\mathfrak{s}}\right)^{2\alpha-1}$, while for $\mathfrak{s} \geq \frac{d}{a^{1/\alpha}\varepsilon}$, the improvement is by a factor of roughly $\frac{1}{a^{\frac{1-\alpha}{\alpha}}\varepsilon^{1-\alpha}}$ (again, ignoring logarithmic factors in both cases). In particular, for *any* $\alpha \in (0, 1)$, when $\mathfrak{s} < \infty$, the asymptotic dependence on ε satisfies $\Lambda_{\mathrm{TN}(a,\alpha)}(\varepsilon, \cdot) = \tilde{\Theta}\big(\varepsilon^{\alpha}\mathcal{M}_{\mathrm{TN}(a,\alpha)}(\varepsilon, \cdot)\big)$, and when $\mathfrak{s} = \infty$, the asymptotic dependence on ε satisfies $\Lambda_{\mathrm{TN}(a,\alpha)}(\varepsilon, \cdot) = \tilde{\Theta}\big(\varepsilon^{\min\{\alpha, 1-\alpha\}}\mathcal{M}_{\mathrm{TN}(a,\alpha)}(\varepsilon, \cdot)\big)$. In either case, we have that for any $\alpha \in (0, 1)$, $\Lambda_{\mathrm{TN}(a,\alpha)}(\varepsilon, \cdot) = o(\mathcal{M}_{\mathrm{TN}(a,\alpha)}(\varepsilon, \cdot))$.

For the Bernstein class condition, the gaps in the upper and lower bounds of Theorem 6 render unclear the necessary and sufficient conditions for $\Lambda_{\mathrm{BC}(a,\alpha)}(\varepsilon, \cdot) = o(\mathcal{M}_{\mathrm{BC}(a,\alpha)}(\varepsilon, \cdot))$. Certainly $\mathfrak{s} < \infty$ is a sufficient condition for this, in which case the improvements are by a

factor of roughly $\frac{1}{a\varepsilon^{\alpha}}$. However, in the case of $\mathfrak{s} = \infty$, the upper bounds do not reveal any improvements over those given above for $\mathcal{M}_{\mathrm{BC}(a,\alpha)}(\varepsilon,\delta)$. Indeed, the example given above in Section 5 reveals that, in some nontrivial cases, $\Lambda_{\mathrm{BC}(a,\alpha)}(\varepsilon,\delta) \gtrsim \mathcal{M}_{\mathrm{BC}(a,\alpha)}(\varepsilon,\delta)/\mathrm{Log}(1/\varepsilon)$, in which case any improvements would be, at best, in the constant and logarithmic factors. Note that this example also presents an interesting contrast between active and passive learning, since it indicates that in some cases $\Lambda_{\mathrm{BC}(a,\alpha)}(\varepsilon,\delta)$ and $\Lambda_{\mathrm{TN}(a,\alpha)}(\varepsilon,\delta)$ are quite different, while the above bounds for passive learning reveal that $\mathcal{M}_{\mathrm{BC}(a,\alpha)}(\varepsilon,\delta)$ is equivalent to $\mathcal{M}_{\mathrm{TN}(a,\alpha)}(\varepsilon,\delta)$ up to constant and logarithmic factors.

In the case of benign noise, comparing the above bounds for passive learning to Theorem 7, we see that (aside from logarithmic factors) the upper bound for active learning improves over the lower bound for passive learning by a factor of roughly $\frac{1}{\nu}$ when $\nu \geq \sqrt{\varepsilon}$. When $\nu < \sqrt{\varepsilon}$, if $\mathfrak{s} > \frac{d}{\varepsilon}$, the improvements are by a factor of roughly $\frac{\nu+\varepsilon}{\varepsilon}$, and if $\mathfrak{s} \leq \frac{d}{\varepsilon}$, the improvements are by roughly a factor of $\min\left\{\frac{1}{\nu}, \frac{(\nu+\varepsilon)d}{\varepsilon^2\mathfrak{s}}\right\}$ (again, ignoring logarithmic factors). However, as has been known for this noise model for some time (Kääriäinen, 2006), there are no gains in terms of the asymptotic dependence on ε for fixed ν. However, if we consider ν_ε such that $\varepsilon \leq \nu_\varepsilon = o(1)$, then for $\mathfrak{s} < \infty$ we have $\Lambda_{\mathrm{BE}(\nu_\varepsilon)}(\varepsilon,\cdot) = \tilde{\Theta}(\nu_\varepsilon \mathcal{M}_{\mathrm{BE}(\nu_\varepsilon)}(\varepsilon,\cdot))$, and for $\mathfrak{s} = \infty$ we have $\Lambda_{\mathrm{BE}(\nu_\varepsilon)}(\varepsilon,\cdot) = \tilde{O}\left(\max\left\{\nu_\varepsilon, \frac{\varepsilon}{\nu_\varepsilon}\right\}\mathcal{M}_{\mathrm{BE}(\nu_\varepsilon)}(\varepsilon,\cdot)\right)$.

Finally, for agnostic noise, similarly to the Bernstein class condition, the gaps between the upper and lower bounds in Theorem 8 render unclear precisely what types of improvements we can expect when $\mathfrak{s} > \frac{1}{\nu+\varepsilon}$, ranging from the lower bound, which has the behavior described above for $\Lambda_{\mathrm{BE}(\nu)}$, to the upper bound, which reflects no improvements over passive learning in this case. When $\mathfrak{s} < \frac{1}{\nu+\varepsilon}$, the upper bound for active learning reflects an improvement over the lower bound for passive learning by roughly a factor of $\frac{1}{(\nu+\varepsilon)\mathfrak{s}}$ (aside from logarithmic factors). It remains an interesting open problem to determine whether the stronger improvements observed for benign noise generally also hold for agnostic noise.

We conclude this section with a remark on the logarithmic factors in the above upper bounds. It is known that the terms of the form "$d\mathrm{Log}(x)$" in each of the above upper bounds for passive learning can be refined to replace x with the maximum of the disagreement coefficient (see Section 7.1 below) over the distributions in $\mathbb{D}$ (Giné and Koltchinskii, 2006; Hanneke and Yang, 2012; Hanneke, 2014). Therefore, based on the results in Section 7.1 relating the disagreement coefficient to the star number, we can replace these "$d\mathrm{Log}(x)$" terms with "$d\mathrm{Log}(\mathfrak{s} \wedge x)$". In the case of $\mathrm{BN}(\beta)$, Massart and Nédélec (2006) and Raginsky and Rakhlin (2011) have argued that, at least in some cases, this logarithmic factor can also be included in the lower bounds. It is presently not known whether this is the case for the other noise models studied here.

7. Connections to the Prior Literature on Active Learning

As mentioned, there is already a substantial literature bounding the label complexities of various active learning algorithms under various noise models. It is natural to ask how the results in the prior literature compare to those stated above. However, as most of the prior results are $\mathcal{P}_{XY}$-dependent, the appropriate comparison is to the worst-case values of those results: that is, maximizing the bounds over $\mathcal{P}_{XY}$ in the respective noise model. This section makes this comparison. In particular, we will see that the label complexity upper

bounds above for RE, BN(β), TN(a, α), and BE(ν) all show some improvements over the known results, with the last two of these showing the strongest improvements.

The general results in the prior literature each express their label complexity bounds in terms of some kind of complexity measure. There are now several such complexity measures in use, each appropriate for studying some family of active learning algorithms under certain noise models. Most of these quantities are dependent on the distribution $\mathcal{P}_{XY}$ or the data, and their definitions are quite diverse. For some pairs of them, there are known inequalities loosely relating them, while other pairs have defied attempts to formally relate the quantities. The dependence on $\mathcal{P}_{XY}$ in the general results in the prior literature is typically isolated to the various complexity measures they are expressed in terms of. Thus, the natural first step is to characterize the worst-case values of these complexity measures, for any given hypothesis class $\mathbb{C}$. Plugging these worst-case values into the original bounds then allows us to compare to the results stated above.

In the process of studying the worst-case behaviors of these complexity measures, we also identify a *very* interesting fact that has heretofore gone unnoticed: namely, that almost all of the complexity measures in the relevant prior literature on the label complexity of active learning are in fact *equal* to the star number when maximized over the choice of distribution or data set. In some sense, this fact is quite surprising, as this seemingly-eclectic collection of complexity measures includes disparate definitions and interpretations, corresponding to entirely distinct approaches to the analysis of the respective algorithms these quantities are used to bound the label complexities of. Thus, this equivalence is interesting in its own right; additionally, it plays an important role in our proofs of the main results above, since it allows us to build on these diverse techniques from the prior literature when establishing these results.

Each subsection below is devoted to a particular complexity measure from the prior literature on active learning, each representing an established technique for obtaining label complexity bounds. Together, they represent a summary of the best-known general results from the prior literature relevant to our present discussion. In each case, we show the equivalence of the worst-case value of the complexity measure to the star number, and then combine this fact with the known results to obtain the corresponding bounds on the minimax label complexities implicit in the prior literature. In each case, we then compare this result to those obtained above.

We additionally study the *doubling dimension*, a quantity which has been used to bound the sample complexity of passive learning, and can be used to provide a loose bound on the label complexity of certain active learning algorithms. Below we argue that, when maximized over the choice of distribution, the doubling dimension can be upper and lower bounded in terms of the star number. One immediate implication of these bounds is that the doubling dimension is bounded if and only if the star number is finite.

Our findings on the relations of these various complexity measures to the star number are summarized in Table 1.

7.1 The Disagreement Coefficient

We begin with, what is perhaps the most well-studied complexity measure in the active learning literature: the *disagreement coefficient* (Hanneke, 2007b, 2009b).

Technique	Source	Relation to $\mathfrak{s}$
disagreement coefficient	(Hanneke, 2007b)	$\sup_P \theta_P(\varepsilon) = \mathfrak{s} \wedge \frac{1}{\varepsilon}$
splitting index	(Dasgupta, 2005)	$\sup_{h,P} \lim_{\tau \to 0} \left\lfloor \frac{1}{\rho_{h,P}(\varepsilon;\tau)} \right\rfloor = \mathfrak{s} \wedge \left\lfloor \frac{1}{\varepsilon} \right\rfloor$
teaching dimension	(Hanneke, 2007a)	$\mathrm{XTD}(\mathbb{C}, m) = \mathfrak{s} \wedge m$
version space compression	(El-Yaniv and Wiener, 2010)	$\max_{h \in \mathbb{C}} \max_{U \in \mathcal{X}^m} \hat{n}_h(\mathcal{U}) = \mathfrak{s} \wedge m$
doubling dimension	(Li and Long, 2007)	$\sup_{h,P} D_{h,P}(\varepsilon) \in [1, O(d)] \log\left(\mathfrak{s} \wedge \frac{1}{\varepsilon}\right)$

Table 1: Many complexity measures from the literature are related to the star number.

Definition 9 *For any $r_0 \geq 0$, any classifier h, and any probability measure $\mathcal{P}$ over $\mathcal{X}$, the disagreement coefficient of h with respect to $\mathbb{C}$ under $\mathcal{P}$ is defined as*

$$\theta_{h,\mathcal{P}}(r_0) = \sup_{r > r_0} \frac{\mathcal{P}\left(\mathrm{DIS}\left(\mathrm{B}_{\mathcal{P}}\left(h, r\right)\right)\right)}{r} \vee 1.$$

Also, for any probability measure $\mathcal{P}_{XY}$ over $\mathcal{X} \times \mathcal{Y}$, letting $\mathcal{P}$ denote the marginal distribution of $\mathcal{P}_{XY}$ over $\mathcal{X}$, and letting $h^*_{\mathcal{P}_{XY}}$ denote a classifier with $\mathrm{er}_{\mathcal{P}_{XY}}(h^*_{\mathcal{P}_{XY}}) = \inf_{h \in \mathbb{C}} \mathrm{er}_{\mathcal{P}_{XY}}(h)$ and $\inf_{h \in \mathbb{C}} \mathcal{P}(x : h(x) \neq h^*_{\mathcal{P}_{XY}}(x)) = 0$,[9] define the disagreement coefficient of the class $\mathbb{C}$ with respect to $\mathcal{P}_{XY}$ as $\theta_{\mathcal{P}_{XY}}(r_0) = \theta_{h^*_{\mathcal{P}_{XY}},\mathcal{P}}(r_0)$.

The disagreement coefficient is used to bound the label complexities of a family of active learning algorithms, described as *disagreement-based*. This line of work was initiated by Cohn, Atlas, and Ladner (1994), who propose an algorithm effective in the realizable case. That method was extended to be robust to label noise by Balcan, Beygelzimer, and Langford (2006, 2009), which then inspired a slew of papers studying variants of this idea; the interested reader is referred to Hanneke (2014) for a thorough survey of this literature. The general-case label complexity analysis of disagreement-based active learning (in terms of the disagreement coefficient) was initiated in the work of Hanneke (2007b, 2009b), and followed up by many papers since then (e.g., Dasgupta, Hsu, and Monteleoni, 2007; Hanneke, 2009a, 2011, 2012; Koltchinskii, 2010; Hanneke and Yang, 2012), as well as many works characterizing the value of the disagreement coefficient under various conditions (e.g., Hanneke, 2007b; Friedman, 2009; Balcan, Hanneke, and Vaughan, 2010; Wang, 2011; Balcan and Long, 2013; Hanneke, 2014); again, see Hanneke (2014) for a thorough survey of the known results on the disagreement coefficient.

To study the worst-case values of the label complexity bounds expressed in terms of the disagreement coefficient, let us define

$$\hat{\hat{\theta}}(\varepsilon) = \sup_{\mathcal{P}_{XY}} \theta_{\mathcal{P}_{XY}}(\varepsilon).$$

In fact, a result of Hanneke (2014, Theorem 7.4) implies that $\hat{\hat{\theta}}(\varepsilon) = \sup_{\mathcal{P}} \sup_{h \in \mathbb{C}} \theta_{h,\mathcal{P}}(\varepsilon)$, so that this would be an equivalent way to define $\hat{\hat{\theta}}(\varepsilon)$, which can sometimes be simpler to

9. See Hanneke (2012) for a proof that such a classifier always exists (though not necessarily in $\mathbb{C}$).

work with. We can now express the bounds on the minimax label complexity implied by the best general results to date in the prior literature on disagreement-based active learning (namely, the results of Hanneke, 2011; Dasgupta, Hsu, and Monteleoni, 2007; Koltchinskii, 2010; Hanneke and Yang, 2012; Hanneke, 2014), summarized as follows (see the survey of Hanneke, 2014, for detailed descriptions of the best-known logarithmic factors in these results).

- $\Lambda_{\mathrm{RE}}(\varepsilon,\delta) \lesssim \hat{\hat{\theta}}(\varepsilon) d \cdot \mathrm{polylog}\left(\frac{1}{\varepsilon\delta}\right)$.
- $\Lambda_{\mathrm{BN}(\beta)}(\varepsilon,\delta) \lesssim \frac{1}{(1-2\beta)^2} \hat{\hat{\theta}}(\varepsilon/(1-2\beta)) d \cdot \mathrm{polylog}\left(\frac{1}{\varepsilon\delta}\right)$.
- $\Lambda_{\mathrm{TN}(a,\alpha)}(\varepsilon,\delta) \lesssim a^2 \left(\frac{1}{\varepsilon}\right)^{2-2\alpha} \hat{\hat{\theta}}(a\varepsilon^\alpha) d \cdot \mathrm{polylog}\left(\frac{1}{\varepsilon\delta}\right)$.
- $\Lambda_{\mathrm{BC}(a,\alpha)}(\varepsilon,\delta) \lesssim a^2 \left(\frac{1}{\varepsilon}\right)^{2-2\alpha} \hat{\hat{\theta}}(a\varepsilon^\alpha) d \cdot \mathrm{polylog}\left(\frac{1}{\varepsilon\delta}\right)$.
- $\Lambda_{\mathrm{BE}(\nu)}(\varepsilon,\delta) \lesssim \left(\frac{\nu^2}{\varepsilon^2}+1\right) \hat{\hat{\theta}}(\nu+\varepsilon) d \cdot \mathrm{polylog}\left(\frac{1}{\varepsilon\delta}\right)$.
- $\Lambda_{\mathrm{AG}(\nu)}(\varepsilon,\delta) \lesssim \left(\frac{\nu^2}{\varepsilon^2}+1\right) \hat{\hat{\theta}}(\nu+\varepsilon) d \cdot \mathrm{polylog}\left(\frac{1}{\varepsilon\delta}\right)$.

In particular, these bounds on $\Lambda_{\mathrm{TN}(a,\alpha)}(\varepsilon,\delta)$, $\Lambda_{\mathrm{BC}(a,\alpha)}(\varepsilon,\delta)$, $\Lambda_{\mathrm{BE}(\nu)}(\varepsilon,\delta)$, and $\Lambda_{\mathrm{AG}(\nu)}(\varepsilon,\delta)$ are the best general-case bounds on the label complexity of active learning in the prior literature (up to logarithmic factors), so that any improvements over these should be considered an interesting advance in our understanding of the capabilities of active learning methods. To compare these results to those stated in Section 5, we need to relate $\hat{\hat{\theta}}(\varepsilon)$ to the star number. Interestingly, we find that these quantities are *equal* (for $\varepsilon = 0$). Specifically, the following result describes the relation between these two quantities; its proof is included in Appendix C.1. This connection also plays a role in the proofs of some of our results from Section 5.

Theorem 10 *$\forall \varepsilon \in (0,1]$, $\hat{\hat{\theta}}(\varepsilon) = \mathfrak{s} \wedge \frac{1}{\varepsilon}$ and $\hat{\hat{\theta}}(0) = \mathfrak{s}$.*

With this result in hand, we immediately observe that several of the upper bounds from Section 5 offer refinements over those stated in terms of $\hat{\hat{\theta}}(\cdot)$ above. For simplicity, we do not discuss differences in the logarithmic factors here. Specifically, the upper bound on $\Lambda_{\mathrm{RE}}(\varepsilon,\delta)$ in Theorem 3 refines that stated here by replacing the factor $\hat{\hat{\theta}}(\varepsilon)d = \min\left\{\mathfrak{s}d, \frac{d}{\varepsilon}\right\}$ with the sometimes-smaller factor $\min\left\{\mathfrak{s}, \frac{d}{\varepsilon}\right\}$. Likewise, the upper bound on $\Lambda_{\mathrm{BN}(\beta)}(\varepsilon,\delta)$ in Theorem 4 refines the result stated here, again by replacing the factor $\hat{\hat{\theta}}(\varepsilon/(1-2\beta))d = \min\left\{\mathfrak{s}d, \frac{(1-2\beta)d}{\varepsilon}\right\}$ with the sometimes-smaller factor $\min\left\{\mathfrak{s}, \frac{(1-2\beta)d}{\varepsilon}\right\}$. On the other hand, Theorem 5 offers a much stronger refinement over the result stated above. Specifically, in the case $\alpha \leq 1/2$, the upper bound in Theorem 5 completely *eliminates* the factor of $\hat{\hat{\theta}}(a\varepsilon^\alpha)$ from the upper bound on $\Lambda_{\mathrm{TN}(a,\alpha)}(\varepsilon,\delta)$ stated here (i.e., replacing it with a universal constant). For the case $\alpha > 1/2$, the upper bound on $\Lambda_{\mathrm{TN}(a,\alpha)}(\varepsilon,\delta)$ in Theorem 5 replaces this factor of $\hat{\hat{\theta}}(a\varepsilon^\alpha) = \min\left\{\mathfrak{s}, \frac{1}{a\varepsilon^\alpha}\right\}$ with the factor $\min\left\{\frac{\mathfrak{s}}{d}, \frac{1}{a^{1/\alpha}\varepsilon}\right\}^{2\alpha-1}$, which is always smaller (for small ε and large d). The upper bounds on $\Lambda_{\mathrm{BC}(a,\alpha)}(\varepsilon,\delta)$ and $\Lambda_{\mathrm{AG}(\nu)}(\varepsilon,\delta)$ in

Theorems 6 and 8 are equivalent to those stated here; indeed, this is precisely how these results are obtained in Appendix B. We have conjectured above that at least the dependence on d and $\mathfrak{s}$ can be refined, analogous to the refinements for the realizable case and bounded noise noted above. However, we *do* obtain refinements for the bound on $\Lambda_{\mathrm{BE}(\nu)}(\varepsilon,\delta)$ in Theorem 7, replacing the factor of $\left(\frac{\nu^2}{\varepsilon^2}+1\right)\hat{\tilde{\theta}}(\nu+\varepsilon)d = \left(\frac{\nu^2}{\varepsilon^2}+1\right)\min\left\{\mathfrak{s}d, \frac{d}{\nu+\varepsilon}\right\}$ in the upper bound here with a factor $\frac{\nu^2}{\varepsilon^2}d + \min\left\{\mathfrak{s}, \frac{d}{\varepsilon}\right\}$, which is sometimes significantly smaller (for $\varepsilon \ll \nu \ll 1$ and large d).

7.2 The Splitting Index

Another, very different, approach to the design and analysis of active learning algorithms was proposed by Dasgupta (2005): namely, the *splitting* approach. In particular, this technique has the desirable property that it yields distribution-dependent label complexity bounds for the realizable case which, even when the marginal distribution $\mathcal{P}$ is held fixed, (almost) imply near-minimax performance. The intuition behind this technique is that the objective in the realizable case (achieving error rate at most ε) is typically well-approximated by the related objective of reducing the *diameter* of the version space (set of classifiers consistent with the observed labels) to size at most ε. From this perspective, at any given time, the impediments to achieving this objective are clearly identifiable: pairs of classifiers $\{h,g\}$ in $\mathbb{C}$ consistent with all labels observed thus far, yet with $\mathcal{P}(x : h(x) \neq g(x)) > \varepsilon$. Supposing we have only a finite number of such classifiers (which can be obtained if we first replace $\mathbb{C}$ by a fine-grained finite *cover* of $\mathbb{C}$), we can then estimate the *usefulness* of a given point X_i by the number of these pairs it would be guaranteed to eliminate if we were to request its label (supposing the worse of the two possible labels); by "eliminate," we mean that at least one of the two classifiers will be inconsistent with the observed label. If we always request labels of points guaranteed to eliminate a large fraction of the surviving ε-separated pairs, we will quickly arrive at a version space of diameter ε, and can then return any surviving classifier. Dasgupta (2005) further applies this strategy in tiers, first eliminating at least one classifier from every $\frac{1}{2}$-separated pair, then repeating this for the remaining $\frac{1}{4}$-separated pairs, and so on. This allows the label complexity to be *localized*, in the sense that the surviving Δ-separated pairs we need to eliminate will be composed of classifiers within distance 2Δ of $f^\star_{\mathcal{P}_{XY}}$ (or the representative thereof in the initial finite cover of $\mathbb{C}$). The analysis of this method naturally leads to the following definition from Dasgupta (2005).

For any finite set $Q \subseteq \{\{h,g\} : h,g \in \mathbb{C}\}$ of unordered pairs of classifiers in $\mathbb{C}$, for any $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, let $Q_x^y = \{\{h,g\} \in Q : h(x) = g(x) = y\}$, and define

$$\mathrm{Split}(Q,x) = |Q| - \max_{y\in\mathcal{Y}} |Q_x^y|.$$

This represents the number of pairs guaranteed to be eliminated (as described above) by requesting the label at a point x. The splitting index is then defined as follows.

Definition 11 *For any $\rho,\Delta,\tau \in [0,1]$, a set $\mathcal{H} \subseteq \mathbb{C}$ is said to be (ρ,Δ,τ)-splittable under a probability measure $\mathcal{P}$ over $\mathcal{X}$ if, for all finite $Q \subseteq \{\{h,g\} \subseteq \mathcal{H} : \mathcal{P}(x : h(x) \neq g(x)) \geq \Delta\}$,*

$$\mathcal{P}(x : \mathrm{Split}(Q,x) \geq \rho|Q|) \geq \tau.$$

For any classifier $h : \mathcal{X} \to \mathcal{Y}$*, any probability measure* $\mathcal{P}$ *over* $\mathcal{X}$*, and any* $\varepsilon, \tau \in [0,1]$*, the* splitting index *is defined as*

$$\rho_{h,\mathcal{P}}(\varepsilon;\tau) = \sup\left\{\rho \in [0,1] : \forall \Delta \geq \varepsilon, \mathrm{B}_{\mathcal{P}}(h, 4\Delta) \text{ is } (\rho,\Delta,\tau)\text{-splittable under } \mathcal{P}\right\}.$$

Dasgupta (2005) proves a bound on the label complexity of a general active learning algorithm based on the above strategy, in the realizable case, expressed in terms of the splitting index. Specifically, for any $\tau > 0$, letting $\rho = \rho_{f^\star_{\mathcal{P}_{XY}},\mathcal{P}}(\varepsilon/4;\tau)$, Dasgupta (2005) finds that for that algorithm to achieve error rate at most ε with probability at least $1-\delta$, it suffices to use a number of label requests

$$\frac{d}{\rho}\mathrm{polylog}\left(\frac{d}{\varepsilon\delta\tau\rho}\right). \tag{1}$$

The τ argument to $\rho_{h,\mathcal{P}}(\varepsilon;\tau)$ captures the trade-off between the number of label requests and the number of unlabeled samples available, with smaller τ corresponding to the scenario where more unlabeled data are available, and a larger value of $\rho_{h,\mathcal{P}}(\varepsilon;\tau)$. Specifically, Dasgupta (2005) argues that $\tilde{O}\left(\frac{d}{\tau\rho}\right)$ unlabeled samples suffice to achieve the above result. In our present model, we suppose an abundance of unlabeled data, and as such, we are interested in the behavior for very small τ. However, note that the logarithmic factors in the above bound have an inverse dependence on τ, so that taking τ too small can potentially increase the value of the bound. It is not presently known whether or not this is necessary (though intuitively it seems not to be). However, for the purpose of comparison to our results in Section 5, we will ignore this logarithmic dependence on $1/\tau$, and focus on the leading factor. In this case, we are interested in the value $\lim_{\tau\to 0}\rho_{h,\mathcal{P}}(\varepsilon;\tau)$. Additionally, to convert (1) into a distribution-free bound for the purpose of comparison to the results in Section 5, we should minimize this value over the choice of $\mathcal{P}$ and $h \in \mathbb{C}$. Formally, we are interested in the following quantity, defined for any $\varepsilon \in [0,1]$.

$$\hat{\hat{\rho}}(\varepsilon) = \inf_P \inf_{h\in\mathbb{C}} \lim_{\tau\to 0} \rho_{h,P}(\varepsilon;\tau).$$

In particular, in terms of this quantity, the maximum possible value of the bound (1) for a given hypothesis class $\mathbb{C}$ is at least

$$\frac{d}{\hat{\hat{\rho}}(\varepsilon/4)}\mathrm{polylog}\left(\frac{d}{\varepsilon\delta}\right).$$

To compare this to the upper bound in Theorem 3, we need to relate $\frac{1}{\hat{\hat{\rho}}(\varepsilon)}$ to the star number. Again, we find that these quantities are essentially *equal* (as $\varepsilon \to 0$), as stated in the following theorem.

Theorem 12 $\forall \varepsilon \in (0,1]$, $\left\lfloor \frac{1}{\hat{\hat{\rho}}(\varepsilon)} \right\rfloor = \mathfrak{s} \wedge \left\lfloor \frac{1}{\varepsilon} \right\rfloor$.

The proof of this result is included in Appendix C.2. We note that the inequalities $\mathfrak{s} \wedge \left\lfloor \frac{1}{\varepsilon} \right\rfloor \leq \left\lfloor \frac{1}{\hat{\hat{\rho}}(\varepsilon)} \right\rfloor \leq \left\lfloor \frac{1}{\varepsilon} \right\rfloor$ were already implicit in the original work of Dasgupta (2005, Corollary 3 and Lemma 1). For completeness (and to make the connection explicit), we

include these arguments in the proof given in Appendix C.2, along with our proof that $\left\lfloor \frac{1}{\hat{\rho}(\varepsilon)} \right\rfloor \leq \mathfrak{s}$ (which was heretofore unknown).

Plugging this into the above bound, we see that the maximum possible value of the bound (1) for a given hypothesis class $\mathbb{C}$ is at least

$$\min\left\{\mathfrak{s}d, \frac{d}{\varepsilon}\right\} \operatorname{polylog}\left(\frac{d}{\varepsilon\delta}\right).$$

Note that the upper bound in Theorem 3 refines this by reducing the first term in the "min" from $\mathfrak{s}d$ to simply $\mathfrak{s}$.

Dasgupta (2005) also argues for a kind of lower bound in terms of the splitting index, which was reformulated as a lower bound on the minimax label complexity (for a fixed $\mathcal{P}$) in the realizable case by Balcan and Hanneke (2012); Hanneke (2014). In our present distribution-free style of analysis, the implication of that result is the following lower bound.

$$\Lambda_{\mathrm{RE}}(\varepsilon, \delta) \gtrsim \frac{1}{\hat{\rho}(4\varepsilon)}.$$

Based on Theorem 12, we see that the $\min\left\{\mathfrak{s}, \frac{1}{\varepsilon}\right\}$ term in the lower bound of Theorem 3 follows immediately from this lower bound. For completeness, in Appendix B, we directly prove this term in the lower bound, based on a more-direct argument than that used to establish the above lower bound. We note, however, that Dasgupta (2005, Corollary 3) also describes a technique for obtaining lower bounds, which is essentially equivalent to that used in Appendix B to obtain this term (and furthermore, makes use of a distribution-dependent version of the "star" idea).

The upper bounds of Dasgupta (2005) have also been extended to the bounded noise setting. In particular, Balcan and Hanneke (2012) and Hanneke (2014) have proposed variants of the splitting approach, which are robust to bounded noise. They have additionally bounded the label complexities of these methods in terms of the splitting index. Similarly to the above discussion of the realizable case, the worst-case values of these bounds for any given hypothesis class $\mathbb{C}$ are larger than those stated in Theorem 4 by factors related to the VC dimension (logarithmic factors aside). We refer the interested readers to these sources for the details of those bounds.

7.3 The Teaching Dimension

Another quantity that has been used to bound the label complexity of certain active learning methods is the *extended teaching dimension growth function*. This quantity was introduced by Hanneke (2007a), inspired by analogous notions used to tightly-characterize the query complexity of *Exact* learning with membership queries (Hegedüs, 1995; Hellerstein, Pillaipakkamnatt, Raghavan, and Wilkins, 1996). The term *teaching dimension* takes its name from the literature on Exact teaching (Goldman and Kearns, 1995), where the teaching dimension characterizes the minimum number of well-chosen labeled data points sufficient to guarantee that the only classifier in $\mathbb{C}$ consistent with these labels is the target function. Hegedüs (1995) extends this to target functions not contained in $\mathbb{C}$, in which case the objective is simply to leave at most one consistent classifier in $\mathbb{C}$; he refers to the minimum number of points sufficient to achieve this as the *extended teaching dimension*, and argues

that this quantity can be used to characterize the minimum number of *membership queries* by a learning algorithm sufficient to guarantee that the only classifier in $\mathbb{C}$ consistent with the returned labels is the target function (which is the objective in *Exact* learning).

Hanneke (2007a) transfers this strategy to the statistical setting studied here (where the objective is only to obtain excess error rate ε with probability $1-\delta$, rather than exactly identifying a target function). That work introduces empirical versions of the teaching dimension and extended teaching dimension, and defines distribution-dependent bounds on these quantities. It then proves upper and lower bounds on the label complexity in terms of these quantities. For our present purposes, we will be most-interested in a particular distribution-free upper bound on these quantities, called the *extended teaching dimension growth function*, also introduced by Hanneke (2006, 2007a). Since both this quantity and the star number are distribution-free, they can be directly compared.

We introduce these quantities formally as follows. For any $m \in \mathbb{N} \cup \{0\}$ and $S \in \mathcal{X}^m$, and for any $h : \mathcal{X} \to \mathcal{Y}$, define the *version space* $V_{S,h} = \{g \in \mathbb{C} : \forall x \in S, g(x) = h(x)\}$ (Mitchell, 1977). For any $m \in \mathbb{N}$ and $\mathcal{U} \in \mathcal{X}^m$, let $\mathbb{C}[\mathcal{U}]$ denote an arbitrary subset of classifiers in $\mathbb{C}$ such that, $\forall h \in \mathbb{C}$, $|\mathbb{C}[\mathcal{U}] \cap V_{\mathcal{U},h}| = 1$: that is, $\mathbb{C}[\mathcal{U}]$ contains exactly one classifier from each equivalence class in $\mathbb{C}$ induced by the classifications of $\mathcal{U}$. For any classifier $h : \mathcal{X} \to \mathcal{Y}$, define

$$\mathrm{TD}(h, \mathbb{C}[\mathcal{U}], \mathcal{U}) = \min\{t \in \mathbb{N} \cup \{0\} : \exists S \in \mathcal{U}^t \text{ s.t. } |V_{S,h} \cap \mathbb{C}[\mathcal{U}]| \leq 1\},$$

the *empirical teaching dimension* of h on $\mathcal{U}$ with respect to $\mathbb{C}[\mathcal{U}]$. Any $S \in \bigcup_t \mathcal{U}^t$ with $|V_{S,h} \cap \mathbb{C}[\mathcal{U}]| \leq 1$ is called a *specifying set* for h on $\mathcal{U}$ with respect to $\mathbb{C}[\mathcal{U}]$; thus, $\mathrm{TD}(h, \mathbb{C}[\mathcal{U}], \mathcal{U})$ is the size of a *minimal specifying set* for h on $\mathcal{U}$ with respect to $\mathbb{C}[\mathcal{U}]$. Equivalently, $S \in \bigcup_t \mathcal{U}^t$ is a specifying set for h on $\mathcal{U}$ with respect to $\mathbb{C}[\mathcal{U}]$ if and only if $\mathrm{DIS}(V_{S,h}) \cap \mathcal{U} = \emptyset$. Also define $\mathrm{TD}(h, \mathbb{C}, m) = \max_{\mathcal{U} \in \mathcal{X}^m} \mathrm{TD}(h, \mathbb{C}[\mathcal{U}], \mathcal{U})$, $\mathrm{TD}(\mathbb{C}, m) = \max_{h \in \mathbb{C}} \mathrm{TD}(h, \mathbb{C}, m)$ (the *teaching dimension growth function*), and $\mathrm{XTD}(\mathbb{C}, m) = \max_{h : \mathcal{X} \to \mathcal{Y}} \mathrm{TD}(h, \mathbb{C}, m)$ (the *extended teaching dimension growth function*).

Hanneke (2007a) proves two upper bounds on the label complexity of active learning relevant to our present discussion. They are summarized as follows (see the original source for the precise logarithmic factors).[10]

- $\Lambda_{\mathrm{RE}}(\varepsilon, \delta) \lesssim \mathrm{XTD}\left(\mathbb{C}, \left\lceil \frac{1}{\varepsilon} \right\rceil\right) d \cdot \mathrm{polylog}\left(\frac{d}{\varepsilon\delta}\right)$.
- $\Lambda_{\mathrm{AG}(\nu)}(\varepsilon, \delta) \lesssim \left(\frac{\nu^2}{\varepsilon^2} + 1\right) \mathrm{XTD}\left(\mathbb{C}, \left\lceil \frac{1}{\nu + \varepsilon} \right\rceil\right) d \cdot \mathrm{polylog}\left(\frac{d}{\varepsilon\delta}\right)$.

Since $\mathrm{BE}(\nu) \subseteq \mathrm{AG}(\nu)$, we have the further implication that

$$\Lambda_{\mathrm{BE}(\nu)}(\varepsilon, \delta) \lesssim \left(\frac{\nu^2}{\varepsilon^2} + 1\right) \mathrm{XTD}\left(\mathbb{C}, \left\lceil \frac{1}{\nu + \varepsilon} \right\rceil\right) d \cdot \mathrm{polylog}\left(\frac{d}{\varepsilon\delta}\right).$$

Additionally, by a refined argument of Hegedüs (1995), the ideas of Hanneke (2007a) can be applied (see Hanneke, 2006, 2009b) to show that

$$\Lambda_{\mathrm{RE}}(\varepsilon, \delta) \lesssim \frac{\mathrm{XTD}(\mathbb{C}, \lceil d/\varepsilon \rceil)}{\log_2(\mathrm{XTD}(\mathbb{C}, \lceil d/\varepsilon \rceil))} d \cdot \mathrm{polylog}\left(\frac{d}{\varepsilon\delta}\right).$$

10. Here we have simplified the arguments m to the $\mathrm{XTD}(\mathbb{C}, m)$ instances compared to those of Hanneke (2007a), using monotonicity of $m \mapsto \mathrm{XTD}(\mathbb{C}, m)$, combined with the basic observation that $\mathrm{XTD}(\mathbb{C}, mk) \leq \mathrm{XTD}(\mathbb{C}, m)k$ for any integer $k \geq 1$.

To compare these bounds to the results stated in Section 5, we will need to relate the quantity $\mathrm{XTD}(\mathbb{C}, m)$ to the star number. Although it may not be obvious from a superficial reading of the definitions, we find that these quantities are *exactly equal* (as $m \to \infty$). Thus, the extended teaching dimension growth function is simply an alternative way of referring to the star number (and vice versa), as they define the same quantity.[11] This equivalence is stated formally in the following theorem, the proof of which is included in Appendix C.3.

Theorem 13 $\forall m \in \mathbb{N}$, $\mathrm{XTD}(\mathbb{C}, m) = \mathrm{TD}(\mathbb{C}, m) = \min\{\mathfrak{s}, m\}$.

We note that the inequalities $\min\{\mathfrak{s}, m\} \leq \mathrm{TD}(\mathbb{C}, m) \leq \mathrm{XTD}(\mathbb{C}, m) \leq m$ follow readily from previously-established facts about the teaching dimension. For instance, Fan (2012) notes that the teaching dimension of any class is at least the maximum degree of its one-inclusion graph; applying this fact to $\mathbb{C}[\mathcal{U}]$ and maximizing over the choice of $\mathcal{U} \in \mathcal{X}^m$, this maximum degree becomes $\min\{\mathfrak{s}, m\}$ (by definition of $\mathfrak{s}$). However, the inequality $\mathrm{XTD}(\mathbb{C}, m) \leq \mathfrak{s}$ and the resulting fact that $\mathrm{XTD}(\mathbb{C}, m) = \mathrm{TD}(\mathbb{C}, m)$ are apparently new.

In fact, in the process of proving this theorem, we establish another remarkable fact: that *every* minimal specifying set is a star set. This is stated formally in the following lemma, the proof of which is also included in Appendix C.3.

Lemma 14 *For any $h : \mathcal{X} \to \mathcal{Y}$, $m \in \mathbb{N}$, and $\mathcal{U} \in \mathcal{X}^m$, every minimal specifying set for h on $\mathcal{U}$ with respect to $\mathbb{C}[\mathcal{U}]$ is a star set for $\mathbb{C} \cup \{h\}$ centered at h.*

Using Theorem 13, we can now compare the results above to those in Section 5. For simplicity, we will not discuss the differences in logarithmic factors here. Specifically, Theorem 3 refines these results on $\Lambda_{\mathrm{RE}}(\varepsilon, \delta)$, replacing a factor of $\min\left\{\mathrm{XTD}(\mathbb{C}, \lceil 1/\varepsilon \rceil)d, \frac{\mathrm{XTD}(\mathbb{C}, \lceil d/\varepsilon \rceil)d}{\log(\mathrm{XTD}(\mathbb{C}, \lceil d/\varepsilon \rceil))}\right\}$ $\approx \min\left\{\mathfrak{s}d, \frac{d}{\varepsilon}, \frac{\mathfrak{s}d}{\log(\mathfrak{s})}, \frac{d^2}{\varepsilon \log(d/\varepsilon)}\right\}$ implied by the above results with a factor of $\min\left\{\mathfrak{s}, \frac{d}{\varepsilon}, \frac{\mathfrak{s}d}{\log(\mathfrak{s})}\right\}$, thus reducing the first term in the "min" by a factor of d (though see below, as Wiener, Hanneke, and El-Yaniv, 2015, have already shown this to be possible, directly in terms of $\mathrm{XTD}(\mathbb{C}, m)$). Theorem 13 further reveals that the above bound on $\Lambda_{\mathrm{AG}(\nu)}(\varepsilon, \delta)$ is equivalent (up to logarithmic factors) to that stated in Theorem 8. However, the bound on $\Lambda_{\mathrm{BE}(\nu)}(\varepsilon, \delta)$ in Theorem 7 refines that implied above, replacing a factor $\left(\frac{\nu^2}{\varepsilon^2} + 1\right) \mathrm{XTD}\left(\mathbb{C}, \left\lceil \frac{1}{\nu+\varepsilon} \right\rceil\right) d \approx \left(\frac{\nu^2}{\varepsilon^2} + 1\right) \min\left\{\mathfrak{s}d, \frac{d}{\nu+\varepsilon}\right\}$ with a factor $\frac{\nu^2}{\varepsilon^2} d + \min\left\{\mathfrak{s}, \frac{d}{\varepsilon}\right\}$, which can be significantly smaller for $\varepsilon \ll \nu \ll 1$ and large d.

Hanneke (2006, 2007a) also proves a *lower bound* on the label complexity of active learning in the realizable case, based on the following modification of the extended teaching dimension. For any set $\mathcal{H} \subseteq \mathbb{C}$, classifier $h : \mathcal{X} \to \mathcal{Y}$, $m \in \mathbb{N}$, $\mathcal{U} \in \mathcal{X}^m$, and $\delta \in [0, 1]$, define the *partial teaching dimension* as

$$\mathrm{XPTD}(h, \mathcal{H}[\mathcal{U}], \mathcal{U}, \delta) = \min\{t \in \mathbb{N} \cup \{0\} : \exists S \in \mathcal{U}^t \text{ s.t. } |V_{S,h} \cap \mathcal{H}[\mathcal{U}]| \leq \delta |\mathcal{H}[\mathcal{U}]| + 1\},$$

and let $\mathrm{XPTD}(\mathcal{H}, m, \delta) = \max_{h:\mathcal{X}\to\mathcal{Y}} \max_{\mathcal{U}\in\mathcal{X}^m} \mathrm{XPTD}(h, \mathcal{H}[\mathcal{U}], \mathcal{U}, \delta)$. Hanneke (2006, 2007a) proves

$$\Lambda_{\mathrm{RE}}(\varepsilon, \delta) \geq \max_{\mathcal{H}\subseteq\mathbb{C}} \mathrm{XPTD}\left(\mathcal{H}, \left\lceil \frac{1-\varepsilon}{\varepsilon} \right\rceil, \delta\right).$$

11. In this sense, the star number is not really a *new* quantity to the active learning literature, but rather a simplified definition for the already-familiar extended teaching dimension growth function.

The following result relates this quantity to the star number.

Theorem 15 $\forall m \in \mathbb{N}$, $\forall \delta \in [0, 1/2]$,

$$\lceil (1-2\delta)\min\{\mathfrak{s}, m\}\rceil \leq \max_{\mathcal{H}\subseteq\mathbb{C}} \mathrm{XPTD}(\mathcal{H}, m, \delta) \leq \left\lceil \left(1 - \frac{\delta}{1+\delta}\right)\min\{\mathfrak{s}, m\}\right\rceil.$$

The proof is in Appendix C.3. Note that, combined with the lower bound of Hanneke (2006, 2007a), this immediately implies the part of the lower bound in Theorem 3 involving $\mathfrak{s}$. In Appendix B, we provide a direct proof for this term in the lower bound, based on an argument similar to that of Hanneke (2007a).

7.3.1 The Version Space Compression Set Size

More-recently, El-Yaniv and Wiener (2010, 2012); Wiener, Hanneke, and El-Yaniv (2015) have studied a quantity $\hat{n}_h(\mathcal{U})$ (for a sequence $\mathcal{U} \in \bigcup_m \mathcal{X}^m$ and classifier h), termed the minimal *version space compression set size*, defined as the size of the smallest subsequence $S \subseteq \mathcal{U}$ for which $V_{S,h} = V_{\mathcal{U},h}$.[12]

It is easy to see that, when $h \in \mathbb{C}$, the version space compression set size is equivalent to the empirical teaching dimension: that is, $\forall h \in \mathbb{C}$,

$$\hat{n}_h(\mathcal{U}) = \mathrm{TD}(h, \mathbb{C}[\mathcal{U}], \mathcal{U}).$$

To see this, note that since $|V_{\mathcal{U},h} \cap \mathbb{C}[\mathcal{U}]| = 1$, any $S \subseteq \mathcal{U}$ with $V_{S,h} = V_{\mathcal{U},h}$ has $|V_{S,h} \cap \mathbb{C}[\mathcal{U}]| = 1$, and hence is a specifying set for h on $\mathcal{U}$ with respect to $\mathbb{C}[\mathcal{U}]$. On the other hand, for any $S \subseteq \mathcal{U}$, we (always) have $V_{S,h} \supseteq V_{\mathcal{U},h}$, so that if $|V_{S,h} \cap \mathbb{C}[\mathcal{U}]| \leq 1$, then $V_{S,h} \cap \mathbb{C}[\mathcal{U}] \supseteq V_{\mathcal{U},h} \cap \mathbb{C}[\mathcal{U}]$ and $|V_{S,h} \cap \mathbb{C}[\mathcal{U}]| \geq |V_{\mathcal{U},h} \cap \mathbb{C}[\mathcal{U}]| = 1 \geq |V_{S,h} \cap \mathbb{C}[\mathcal{U}]|$, which together imply $V_{S,h} \cap \mathbb{C}[\mathcal{U}] = V_{\mathcal{U},h} \cap \mathbb{C}[\mathcal{U}]$; thus, $V_{S,h} \subseteq \{g \in \mathbb{C} : \forall x \in \mathcal{U}, g(x) = h(x)\} = V_{\mathcal{U},h} \subseteq V_{S,h}$, so that $V_{S,h} = V_{\mathcal{U},h}$: that is, S is a version space compression set. Thus, in the case $h \in \mathbb{C}$, any version space compression set S is a specifying set for h on $\mathcal{U}$ with respect to $\mathbb{C}[\mathcal{U}]$ and vice versa. That $\hat{n}_h(\mathcal{U}) = \mathrm{TD}(h, \mathbb{C}[\mathcal{U}], \mathcal{U})$ $\forall h \in \mathbb{C}$ follows immediately from this equivalence.

In particular, combined with Theorem 13, this implies that $\forall m \in \mathbb{N}$,

$$\max_{\mathcal{U}\in\mathcal{X}^m} \max_{h\in\mathbb{C}} \hat{n}_h(\mathcal{U}) = \mathrm{TD}(\mathbb{C}, m) = \min\{\mathfrak{s}, m\}. \tag{2}$$

Letting $\hat{n}_m = \hat{n}_{f^\star_{\mathcal{P}_{XY}}}(\{X_1, \ldots, X_m\})$, Wiener, Hanneke, and El-Yaniv (2015) have shown that, in the realizable case, for the CAL active learning algorithm (proposed by Cohn, Atlas, and Ladner, 1994) to achieve error rate at most ε with probability at least $1-\delta$, it suffices to use a budget n of any size at least

$$\max_{1\leq m\leq M_{\varepsilon,\delta}} \hat{n}_m \cdot \mathrm{polylog}\left(\frac{1}{\varepsilon\delta}\right),$$

where $M_{\varepsilon,\delta} \lesssim \frac{1}{\varepsilon}\left(d\mathrm{Log}\left(\frac{1}{\varepsilon}\right) + \mathrm{Log}\left(\frac{1}{\delta}\right)\right)$ is a bound on the sample complexity of passive learning by returning an arbitrary classifier in the version space (Vapnik, 1982, 1998; Blumer,

12. The quantity studied there is defined slightly differently, but is easily seen to be equivalent to this definition.

Ehrenfeucht, Haussler, and Warmuth, 1989). They further provide a distribution-dependent bound (to remove the dependence on the data here) based on confidence bounds on $\hat{n}_m$ (analogous to the aforementioned distribution-dependent bounds on the empirical teaching dimension studied by Hanneke, 2007a). For our purposes (distribution-free, data-independent bounds), we can simply take the maximum over possible data sets and possible $f^\star_{\mathcal{P}_{XY}}$ functions, so that the above bound becomes

$$\max_{x_1,x_2,\ldots\in\mathcal{X}} \max_{h\in\mathbb{C}} \max_{1\le m\le M_{\varepsilon,\delta}} \hat{n}_h(\{x_1,\ldots,x_m\})\text{polylog}\left(\frac{1}{\varepsilon\delta}\right)$$
$$= \text{TD}\left(\mathbb{C}, M_{\varepsilon,\delta}\right)\text{polylog}\left(\frac{1}{\varepsilon\delta}\right) \lesssim \text{TD}\left(\mathbb{C}, \left\lfloor\frac{d}{\varepsilon}\right\rfloor\right)\text{polylog}\left(\frac{1}{\varepsilon\delta}\right).$$

Combining this with (2), we find that the label complexity of CAL in the realizable case is at most

$$\min\left\{\mathfrak{s}, \frac{d}{\varepsilon}\right\}\text{polylog}\left(\frac{1}{\varepsilon\delta}\right),$$

which matches the upper bound on the minimax label complexity from Theorem 3 up to logarithmic factors.

7.4 The Doubling Dimension

Another quantity of interest in the learning theory literature is the *doubling dimension*, also known as the *local metric entropy* (LeCam, 1973; Yang and Barron, 1999; Gupta, Krauthgamer, and Lee, 2003; Bshouty, Li, and Long, 2009). Specifically, for any set $\mathcal{H}$ of classifiers, a set of classifiers $\mathcal{G}$ is an ε-cover of $\mathcal{H}$ (with respect to the $\mathcal{P}(\text{DIS}(\{\cdot,\cdot\}))$ pseudometric) if

$$\sup_{h\in\mathcal{H}}\inf_{g\in\mathcal{G}} \mathcal{P}(x : g(x)\neq h(x)) \le \varepsilon.$$

Let $\mathcal{N}(\varepsilon,\mathcal{H},\mathcal{P})$ denote the minimum cardinality $|\mathcal{G}|$ over all ε-covers $\mathcal{G}$ of $\mathcal{H}$, or else $\mathcal{N}(\varepsilon,\mathcal{H},\mathcal{P}) = \infty$ if no finite ε-cover of $\mathcal{H}$ exists. The doubling dimension (at h) is defined as follows.

Definition 16 *For any $\varepsilon\in(0,1]$, any probability measure P over $\mathcal{X}$, and any classifier h, define*

$$D_{h,P}(\varepsilon) = \max_{r\ge\varepsilon}\log_2\left(\mathcal{N}\left(r/2, \text{B}_P(h,r), P\right)\right).$$

The quantity $D_\varepsilon = D_{f^\star_{\mathcal{P}_{XY}},\mathcal{P}}(\varepsilon)$ is known to be useful in bounding the sample complexity of passive learning. Specifically, Li and Long (2007); Bshouty, Li, and Long (2009) have shown that there is a passive learning algorithm achieving sample complexity $\lesssim \frac{D_{\varepsilon/4}}{\varepsilon} + \frac{1}{\varepsilon}\log\left(\frac{1}{\delta}\right)$ for $\mathcal{P}_{XY}\in\text{RE}$. Furthermore, though we do not go into the details here, by a combination of the ideas from Dasgupta (2005), Balcan, Beygelzimer, and Langford (2009), and Hanneke (2007b), it is possible to show that a certain active learning algorithm achieves a label complexity $\lesssim 4^{D_\varepsilon}D_\varepsilon\cdot\text{polylog}(\frac{1}{\varepsilon\delta})$ for $\mathcal{P}_{XY}\in\text{RE}$, though this is typically a very loose upper bound.

To our knowledge, the question of the worst-case value of the doubling dimension for a given hypothesis class $\mathbb{C}$ has not previously been explored in the literature (though there is

an obvious $O(d \log(1/\varepsilon))$ upper bound derivable from the literature on covering numbers). Here we obtain upper and lower bounds on this worst-case value, expressed in terms of the star number. While this relation generally has a wide range (roughly a factor of d), it does have the interesting implication that the doubling dimension is *bounded* if and only if $\mathfrak{s} < \infty$. Specifically, we have the following theorem, the proof of which is included in Appendix C.4.

Theorem 17 $\forall \varepsilon \in (0, 1/4]$, $\max\left\{d, \text{Log}\left(\mathfrak{s} \wedge \frac{1}{\varepsilon}\right)\right\} \lesssim \sup_P \sup_{h \in \mathbb{C}} D_{h,P}(\varepsilon) \lesssim d\text{Log}\left(\mathfrak{s} \wedge \frac{1}{\varepsilon}\right)$.

One can show that the gap between the upper and lower bounds on $\sup_P \sup_{h \in \mathbb{C}} D_{h,P}(\varepsilon)$ in this result cannot generally be improved by much without sacrificing generality or introducing additional quantities. Specifically, for the class $\mathbb{C}$ discussed in Appendix D.2, we have $\sup_P \sup_{h \in \mathbb{C}} D_{h,P}(\varepsilon) \leq \sup_P \log_2(\mathcal{N}(\varepsilon/2, \mathbb{C}, P)) \lesssim \max\left\{d, \text{Log}\left(\mathfrak{s} \wedge \frac{1}{\varepsilon}\right)\right\}$, so that the lower bound above is sometimes tight to within a universal constant factor. For the class $\mathbb{C}$ discussed in Appendix D.1, based on a result of Raginsky and Rakhlin (2011, Lemma 4), one can show $\sup_P \sup_{h \in \mathbb{C}} D_{h,P}(\varepsilon) \gtrsim d\text{Log}\left(\frac{\mathfrak{s}}{d} \wedge \frac{1}{\varepsilon}\right)$, so that the above upper bound is sometimes tight, aside from a small difference in the logarithmic factor (dividing $\mathfrak{s}$ by d).

Interestingly, in the process of proving the upper bound in Theorem 17, we also establish the following inequality relating the doubling dimension and the disagreement coefficient, holding for any classifier h, any probability measure $\mathcal{P}$ over $\mathcal{X}$, and any $\varepsilon \in (0, 1]$.

$$D_{h,\mathcal{P}}(\varepsilon) \leq 2d \log_2\left(22e^2 \theta_{h,\mathcal{P}}(\varepsilon)\right).$$

This inequality may be of independent interest, as it enables comparisons between results in the literature expressed in terms of these quantities. For instance, it implies that in the realizable case, the passive learning sample complexity bound of Bshouty, Li, and Long (2009) is no larger than that of Giné and Koltchinskii (2006) (aside from constant factors).

8. Conclusions

In this work, we derived upper and lower bounds on the minimax label complexity of active learning under several noise models. In most cases, these new bounds offer refinements over the best results in the prior literature. Furthermore, in the case of Tsybakov noise, we discovered the heretofore-unknown fact that the minimax label complexity of active learning with VC classes is *always* smaller than that of passive learning. We expressed each of these bounds in terms of a simple combinatorial complexity measure, termed the *star number*. We further found that almost all of the distribution-dependent and sample-dependent complexity measures in the prior active learning literature are exactly equal to the star number when maximized over the choice of distribution or data set.

The bounds derived here are all distribution-free, in the sense that they are expressed without dependence or restrictions on the marginal distribution $\mathcal{P}$ over $\mathcal{X}$. They are also worst-case bounds, in the sense that they express the maximum of the label complexity over the distributions in the noise model $\mathbb{D}$, rather than expressing a bound on the label complexity achieved by a given algorithm as a function of $\mathcal{P}_{XY}$. As observed by Dasgupta (2005), there are some cases in which smaller label complexities can be achieved under restrictions on the marginal distribution $\mathcal{P}$, and some cases in which there are achievable

label complexities which exhibit a range of values depending on $\mathcal{P}_{XY}$ (see also Balcan, Hanneke, and Vaughan, 2010; Hanneke, 2012, for further exploration of this). Our results reveal that in some cases, such as Tsybakov noise with $\alpha \leq 1/2$, these issues might typically not be of much significance (aside from logarithmic factors). However, in other cases, particularly when $\mathfrak{s} = \infty$, the issue of expressing distribution-dependent bounds on the label complexity is clearly an important one. In particular, the question of the minimax label complexity of active learning under the restrictions of the above noise models that explicitly fix the marginal distribution $\mathcal{P}$ remains an important and challenging open problem. In deriving such bounds, the present work should be considered a kind of guide, in that we should restrict our focus to deriving distribution-dependent label complexity bounds with worst-case values that are never worse than the distribution-free bounds proven here.

Appendix A. Preliminary Lemmas

Before presenting the proofs of the main results above, we begin by introducing some basic lemmas, which will be useful in the main proofs below.

A.1 ε-nets and ε-covers

For a collection $\mathcal{T}$ of measurable subsets of $\mathcal{X}$, a value $\varepsilon \geq 0$, and a probability measure $\mathcal{P}$ on $\mathcal{X}$, we say a set $N \subseteq \mathcal{X}$ is an ε-net of $\mathcal{P}$ for $\mathcal{T}$ if $N \cap A \neq \emptyset$ for every $A \in \mathcal{T}$ with $\mathcal{P}(A) > \varepsilon$ (Haussler and Welzl, 1987). Also, a finite set $\mathcal{H}$ of classifiers is called an ε-cover of $\mathbb{C}$ (under the $\mathcal{P}(\mathrm{DIS}(\{\cdot,\cdot\}))$ pseudometric) if $\sup_{g\in\mathbb{C}} \min_{h\in\mathcal{H}} \mathcal{P}(x : h(x) \neq g(x)) \leq \varepsilon$.

The following lemma bounds the probabilities and empirical probabilities of sets in a collection in terms of each other. This result is based on the work of Vapnik and Chervonenkis (1974) (see also Vapnik, 1982, Theorem A.3); this version is taken from Bousquet, Boucheron, and Lugosi (2004, Theorem 7), in combination with the VC-Sauer Lemma (Vapnik and Chervonenkis, 1971; Sauer, 1972) and a union bound.

Lemma 18 *For any collection $\mathcal{T}$ of measurable subsets of $\mathcal{X}$, letting k denote the VC dimension of $\mathcal{T}$, for any $\delta \in (0,1)$, for any integer $m > k$, for any probability measure $\mathcal{P}$ over $\mathcal{X}$, if $X_1', \ldots, X_m'$ are independent $\mathcal{P}$-distributed random variables, then with probability at least $1-\delta$, it holds that $\forall A \in \mathcal{T}$, letting $\hat{\mathcal{P}}(A) = \frac{1}{m}\sum_{i=1}^m \mathbb{1}_A(X_i')$,*

$$\mathcal{P}(A) \leq \hat{\mathcal{P}}(A) + 2\sqrt{\mathcal{P}(A)\frac{k\mathrm{Log}\left(\frac{2em}{k}\right) + \mathrm{Log}\left(\frac{8}{\delta}\right)}{m}}$$

$$\text{and } \hat{\mathcal{P}}(A) \leq \mathcal{P}(A) + 2\sqrt{\hat{\mathcal{P}}(A)\frac{k\mathrm{Log}\left(\frac{2em}{k}\right) + \mathrm{Log}\left(\frac{8}{\delta}\right)}{m}}.$$

In particular, with a bit of algebra, this implies the following corollary.

Corollary 19 *There exists a finite universal constant $c_0 \geq 1$ such that, for any collection $\mathcal{T}$ of measurable subsets of $\mathcal{X}$, letting k denote the VC dimension of $\mathcal{T}$, for any $\varepsilon, \delta \in (0,1)$, for any integer $m \geq \frac{c_0}{\varepsilon}\left(k\mathrm{Log}\left(\frac{1}{\varepsilon}\right) + \mathrm{Log}\left(\frac{1}{\delta}\right)\right)$, for any probability measure $\mathcal{P}$ over $\mathcal{X}$, if $X_1', \ldots, X_m'$ are independent $\mathcal{P}$-distributed random variables, then with probability at least $1-\delta$, it holds that $\forall A \in \mathcal{T}$, letting $\hat{\mathcal{P}}(A) = \frac{1}{m}\sum_{i=1}^m \mathbb{1}_A(X_i')$,*

- $\hat{\mathcal{P}}(A) \le \frac{3}{4}\varepsilon \implies \mathcal{P}(A) < \varepsilon$,
- $\mathcal{P}(A) \le \frac{1}{2}\varepsilon \implies \hat{\mathcal{P}}(A) < \frac{3}{4}\varepsilon$.

Proof Let $\mathcal{E}(m) = 4\frac{k\mathrm{Log}\left(\frac{2em}{k}\right)+\mathrm{Log}\left(\frac{8}{\delta}\right)}{m}$, and note that for $m \ge \frac{c_0}{\varepsilon}\left(k\mathrm{Log}\left(\frac{1}{\varepsilon}\right)+\mathrm{Log}\left(\frac{1}{\delta}\right)\right)$,

$$\mathcal{E}(m) \le \frac{4\varepsilon}{c_0}\frac{k\mathrm{Log}\left(\frac{2ec_0}{k\varepsilon}\left(k\mathrm{Log}\left(\frac{1}{\varepsilon}\right)+\mathrm{Log}\left(\frac{1}{\delta}\right)\right)\right)+\mathrm{Log}\left(\frac{8}{\delta}\right)}{k\mathrm{Log}\left(\frac{1}{\varepsilon}\right)+\mathrm{Log}\left(\frac{1}{\delta}\right)} \tag{3}$$

If $k\mathrm{Log}\left(\frac{1}{\varepsilon}\right) \ge \mathrm{Log}\left(\frac{1}{\delta}\right)$, then

$$\begin{aligned}
&k\mathrm{Log}\left(\frac{2ec_0}{k\varepsilon}\left(k\mathrm{Log}\left(\frac{1}{\varepsilon}\right)+\mathrm{Log}\left(\frac{1}{\delta}\right)\right)\right)+\mathrm{Log}\left(\frac{8}{\delta}\right)\\
&\le k\mathrm{Log}\left(\frac{4ec_0}{\varepsilon}\mathrm{Log}\left(\frac{1}{\varepsilon}\right)\right)+\mathrm{Log}\left(\frac{8}{\delta}\right) \le k\mathrm{Log}\left(\frac{4ec_0}{\varepsilon^2}\right)+\mathrm{Log}\left(\frac{8}{\delta}\right)\\
&\le 2k\mathrm{Log}\left(\frac{1}{\varepsilon}\right)+k\mathrm{Log}(4ec_0)+\mathrm{Log}(8)+\mathrm{Log}\left(\frac{1}{\delta}\right) \le \mathrm{Log}\left(32e^3c_0\right)\left(k\mathrm{Log}\left(\frac{1}{\varepsilon}\right)+\mathrm{Log}\left(\frac{1}{\delta}\right)\right).
\end{aligned}$$

Otherwise, if $k\mathrm{Log}\left(\frac{1}{\varepsilon}\right) < \mathrm{Log}\left(\frac{1}{\delta}\right)$, then

$$\begin{aligned}
&k\mathrm{Log}\left(\frac{2ec_0}{k\varepsilon}\left(k\mathrm{Log}\left(\frac{1}{\varepsilon}\right)+\mathrm{Log}\left(\frac{1}{\delta}\right)\right)\right)+\mathrm{Log}\left(\frac{8}{\delta}\right)\\
&\le k\mathrm{Log}\left(\frac{4ec_0}{k\varepsilon}\mathrm{Log}\left(\frac{1}{\delta}\right)\right)+\mathrm{Log}\left(\frac{8}{\delta}\right) \le k\mathrm{Log}\left(\frac{4ec_0}{\varepsilon}\right)+k\mathrm{Log}\left(\frac{1}{k}\mathrm{Log}\left(\frac{1}{\delta}\right)\right)+\mathrm{Log}\left(\frac{8}{\delta}\right),
\end{aligned}$$

and since $x \mapsto x\mathrm{Log}\left(\frac{1}{x}\mathrm{Log}\left(\frac{1}{\delta}\right)\right)$ is nondecreasing for $x > 0$, and $k \le k\mathrm{Log}\left(\frac{1}{\varepsilon}\right) \le \mathrm{Log}\left(\frac{1}{\delta}\right)$, the above is at most

$$\begin{aligned}
&k\mathrm{Log}\left(\frac{4ec_0}{\varepsilon}\right)+\mathrm{Log}\left(\frac{1}{\delta}\right)+\mathrm{Log}\left(\frac{8}{\delta}\right)\\
&\le k\mathrm{Log}\left(\frac{1}{\varepsilon}\right)+k\mathrm{Log}(4ec_0)+\mathrm{Log}(8)+2\mathrm{Log}\left(\frac{1}{\delta}\right) \le \mathrm{Log}\left(32e^2c_0\right)\left(k\mathrm{Log}\left(\frac{1}{\varepsilon}\right)+\mathrm{Log}\left(\frac{1}{\delta}\right)\right).
\end{aligned}$$

In either case, we have that the right hand side of (3) is at most $\frac{4\varepsilon}{c_0}\mathrm{Log}\left(32e^3c_0\right)$. In particular, taking $c_0 = 2^{14}$ suffices to make $\frac{4}{c_0}\mathrm{Log}\left(32e^3c_0\right) \le \frac{1}{64}$, so that (3) implies $\mathcal{E}(m) \le \frac{\varepsilon}{64}$.

Lemma 18 implies that with probability at least $1-\delta$, every $A \in \mathcal{T}$ has

$$\mathcal{P}(A) \le \hat{\mathcal{P}}(A) + \sqrt{\mathcal{P}(A)\mathcal{E}(m)}$$

and

$$\hat{\mathcal{P}}(A) \le \mathcal{P}(A) + \sqrt{\hat{\mathcal{P}}(A)\mathcal{E}(m)}.$$

Solving these quadratic expressions in $\sqrt{\mathcal{P}(A)}$ and $\sqrt{\hat{\mathcal{P}}(A)}$, respectively, we have

$$\mathcal{P}(A) \le \hat{\mathcal{P}}(A) + \frac{1}{2}\mathcal{E}(m) + \frac{1}{2}\sqrt{\mathcal{E}(m)^2 + 4\mathcal{E}(m)\hat{\mathcal{P}}(A)} \tag{4}$$

and

$$\hat{\mathcal{P}}(A) \leq \mathcal{P}(A) + \frac{1}{2}\mathcal{E}(m) + \frac{1}{2}\sqrt{\mathcal{E}(m)^2 + 4\mathcal{E}(m)\mathcal{P}(A)}. \tag{5}$$

Therefore, if $\hat{\mathcal{P}}(A) \leq \frac{3}{4}\varepsilon$, then (4) implies

$$\begin{aligned}\mathcal{P}(A) &\leq \frac{3}{4}\varepsilon + \frac{1}{2}\mathcal{E}(m) + \frac{1}{2}\sqrt{\mathcal{E}(m)^2 + 3\mathcal{E}(m)\varepsilon} \\ &\leq \left(\frac{3}{4} + \frac{1}{128} + \frac{1}{2}\sqrt{\frac{1}{64^2} + \frac{3}{64}}\right)\varepsilon < \left(\frac{3}{4} + \frac{1}{128} + \frac{1}{8}\right)\varepsilon < \varepsilon,\end{aligned}$$

and likewise, if $\mathcal{P}(A) \leq \frac{1}{2}\varepsilon$, then (5) implies

$$\begin{aligned}\hat{\mathcal{P}}(A) &\leq \frac{1}{2}\varepsilon + \frac{1}{2}\mathcal{E}(m) + \frac{1}{2}\sqrt{\mathcal{E}(m)^2 + 2\mathcal{E}(m)\varepsilon} \\ &\leq \left(\frac{1}{2} + \frac{1}{128} + \frac{1}{2}\sqrt{\frac{1}{64^2} + \frac{1}{32}}\right)\varepsilon < \left(\frac{1}{2} + \frac{1}{128} + \frac{1}{8}\right)\varepsilon < \frac{3}{4}\varepsilon.\end{aligned}$$

■

We will be interested in applying these results to the collection of sets $\{\mathrm{DIS}(\{h,g\}) : h,g \in \mathbb{C}\}$. For this, the following lemma of Vidyasagar (2003, Theorem 4.5) will be useful.

Lemma 20 *The VC dimension of the collection* $\{\mathrm{DIS}(\{h,g\}) : h,g \in \mathbb{C}\}$ *is at most* $10d$.

Together, these results imply the following lemma (see also Vapnik and Chervonenkis, 1974; Vapnik, 1982; Blumer, Ehrenfeucht, Haussler, and Warmuth, 1989; Haussler and Welzl, 1987).

Lemma 21 *There exists a finite universal constant* $c \geq 1$ *such that, for any* $\varepsilon, \delta \in (0,1)$, *for any integer* $m \geq \frac{c}{\varepsilon}\left(d\mathrm{Log}\left(\frac{1}{\varepsilon}\right) + \mathrm{Log}\left(\frac{1}{\delta}\right)\right)$, *for any probability measure* $\mathcal{P}$ *over* $\mathcal{X}$, *if* $X_1', \ldots, X_m'$ *are independent* $\mathcal{P}$*-distributed random variables, then with probability at least* $1-\delta$, *it holds that* $\forall h, g \in \mathbb{C}$, *if* $(g(X_1'), \ldots, g(X_m')) = (h(X_1'), \ldots, h(X_m'))$, *then* $\mathcal{P}(x : g(x) \neq h(x)) \leq \varepsilon$.
In particular, this implies that with probability at least $1-\delta$, *letting* $\mathbb{C}[(X_1', \ldots, X_m')]$ *be as in Section 7.3,* $\mathbb{C}[(X_1', \ldots, X_m')]$ *is an* ε*-cover of* $\mathbb{C}$ *(under the* $\mathcal{P}(\mathrm{DIS}(\{\cdot,\cdot\}))$ *pseudometric), and* $\{X_1', \ldots, X_m'\}$ *is an* ε*-net of* $\mathcal{P}$ *for* $\{\mathrm{DIS}(\{h,g\}) : h,g \in \mathbb{C}\}$.

Proof Let c_0 be as in Corollary 19, and let k denote the VC dimension of $\{\mathrm{DIS}(\{h,g\}) : h,g \in \mathbb{C}\}$. Corollary 19 implies that, if $m \geq \frac{c_0}{\varepsilon}\left(k\mathrm{Log}\left(\frac{1}{\varepsilon}\right) + \mathrm{Log}\left(\frac{1}{\delta}\right)\right)$, then there is an event E of probability at least $1-\delta$, on which every $h, g \in \mathbb{C}$ with $\sum_{t=1}^m \mathbb{1}_{\mathrm{DIS}(\{h,g\})}(X_t') = 0$ satisfy $\mathcal{P}(\mathrm{DIS}(\{h,g\})) < \varepsilon$; in particular, this proves that on the event E, $\{X_1', \ldots, X_m'\}$ is an ε-net of $\mathcal{P}$ for $\{\mathrm{DIS}(\{h,g\}) : h,g \in \mathbb{C}\}$. Furthermore, by definition of $\mathbb{C}[(X_1', \ldots, X_m')]$, for every $h \in \mathbb{C}$, $\exists g \in \mathbb{C}[(X_1', \ldots, X_m')]$ with $\sum_{t=1}^m \mathbb{1}_{\mathrm{DIS}(\{h,g\})}(X_t') = 0$, which (on the event E) therefore also satisfies $\mathcal{P}(\mathrm{DIS}(\{h,g\})) < \varepsilon$. Thus, on the event E, $\mathbb{C}[(X_1', \ldots, X_m')]$ is an ε-cover of $\mathbb{C}$ (under the $\mathcal{P}(\mathrm{DIS}(\{\cdot,\cdot\}))$ pseudometric). To complete the proof, we note that Lemma 20 implies $k \leq 10d$, so that by choosing $c = 10c_0$, the condition $m \geq \frac{c_0}{\varepsilon}\left(k\mathrm{Log}\left(\frac{1}{\varepsilon}\right) + \mathrm{Log}\left(\frac{1}{\delta}\right)\right)$

will be satisfied for any $m \geq \frac{c}{\varepsilon}\left(d\mathrm{Log}\left(\frac{1}{\varepsilon}\right) + \mathrm{Log}\left(\frac{1}{\delta}\right)\right)$. ■

Based on this result, it is straightforward to construct an ε-net of $\mathcal{P}$ for $\{\mathrm{DIS}(\{h,g\}) : h,g \in \mathbb{C}\}$ of size $\lesssim \frac{d}{\varepsilon}\mathrm{Log}\left(\frac{1}{\varepsilon}\right)$, based on a relatively small number of random samples. Specifically, we have the following lemma.

Lemma 22 *There exists a finite universal constant $c' \geq 1$ such that, for any probability measure $\mathcal{P}$ on $\mathcal{X}$, if $X'_1, X'_2, \ldots$ are independent $\mathcal{P}$-distributed random variables, then $\forall \varepsilon, \delta \in (0,1)$, for any integers $m \geq \frac{c'd}{\varepsilon}\mathrm{Log}\left(\frac{1}{\varepsilon}\right)$ and $\ell \geq \frac{c'}{\varepsilon}\left(d\mathrm{Log}\left(\frac{1}{\varepsilon}\right) + \mathrm{Log}\left(\frac{1}{\delta}\right)\right)$, defining $N_i = \{X'_{m(i-1)+1}, \ldots, X'_{mi}\}$ for each $i \in \{1, \ldots, \lceil \log_2(2/\delta)\rceil\}$, letting*

$$\hat{i} = \operatorname*{argmin}_{i \in \{1,\ldots,\lceil \log_2(2/\delta)\rceil\}} \max\left\{ \sum_{j=m\lceil \log_2(2/\delta)\rceil+1}^{m\lceil \log_2(2/\delta)\rceil+\ell} \mathbb{1}_{\mathrm{DIS}(\{h,g\})}(X'_j) : h,g \in \mathbb{C}, \sum_{j=m(i-1)+1}^{mi} \mathbb{1}_{\mathrm{DIS}(\{h,g\})}(X'_j) = 0 \right\},$$

and $\hat{N} = N_{\hat{i}}$, with probability at least $1-\delta$, $\hat{N}$ is an ε-net of $\mathcal{P}$ for $\{\mathrm{DIS}(\{h,g\}) : h,g \in \mathbb{C}\}$.

Proof Let k denote the VC dimension of the collection of sets $\{\mathrm{DIS}(\{h,g\}) : h,g \in \mathbb{C}\}$. Letting c_0 be as in Corollary 19, taking $c' \geq 10c_0$, we have $\ell \geq \frac{c_0}{\varepsilon}\left(10d\mathrm{Log}\left(\frac{1}{\varepsilon}\right) + \mathrm{Log}\left(\frac{2}{\delta}\right)\right)$, which is at least $\frac{c_0}{\varepsilon}\left(k\mathrm{Log}\left(\frac{1}{\varepsilon}\right) + \mathrm{Log}\left(\frac{2}{\delta}\right)\right)$ by Lemma 20. Therefore, Corollary 19 implies there exists an event E' of probability at least $1-\delta/2$ such that, on E', $\forall h,g \in \mathbb{C}$,

$$\sum_{j=m\lceil \log_2(2/\delta)\rceil+1}^{m\lceil \log_2(2/\delta)\rceil+\ell} \mathbb{1}_{\mathrm{DIS}(\{h,g\})}(X'_j) \leq \frac{3}{4}\varepsilon\ell \implies \mathcal{P}(\mathrm{DIS}(\{h,g\})) \leq \varepsilon, \tag{6}$$

$$\mathcal{P}(\mathrm{DIS}(\{h,g\})) \leq \frac{\varepsilon}{2} \implies \sum_{j=m\lceil \log_2(2/\delta)\rceil+1}^{m\lceil \log_2(2/\delta)\rceil+\ell} \mathbb{1}_{\mathrm{DIS}(\{h,g\})}(X'_j) \leq \frac{3}{4}\varepsilon\ell. \tag{7}$$

Let c be as in Lemma 21. Taking $c' \geq 6c$, we have $m \geq \frac{2c}{\varepsilon}\left(d\mathrm{Log}\left(\frac{2}{\varepsilon}\right) + \mathrm{Log}\left(2\right)\right)$, so that Lemma 21 implies that, for each $i \in \{1, \ldots, \lceil \log_2(2/\delta)\rceil\}$, N_i is an $\frac{\varepsilon}{2}$-net of $\mathcal{P}$ for $\{\mathrm{DIS}(\{h,g\}) : h,g \in \mathbb{C}\}$ with probability at least $1/2$. Since the N_i sets are independent, there is an event E of probability at least $1-(1-1/2)^{\lceil \log_2(2/\delta)\rceil} \geq 1-\delta/2$, on which $\exists i^* \in \{1, \ldots, \lceil \log_2(2/\delta)\rceil\}$ such that N_{i^*} is an $\frac{\varepsilon}{2}$-net of $\mathcal{P}$ for $\{\mathrm{DIS}(\{h,g\}) : h,g \in \mathbb{C}\}$. In particular, this implies that on E,

$$\sup\left\{ \mathcal{P}(\mathrm{DIS}(\{h,g\})) : h,g \in \mathbb{C}, \sum_{j=m(i^*-1)+1}^{mi^*} \mathbb{1}_{\mathrm{DIS}(\{h,g\})}(X'_j) = 0 \right\} \leq \frac{\varepsilon}{2}. \tag{8}$$

Therefore, on the event $E' \cap E$, we have

$$\max\left\{\sum_{j=m\lceil\log_2(2/\delta)\rceil+1}^{m\lceil\log_2(2/\delta)\rceil+\ell} \mathbb{1}_{\mathrm{DIS}(\{h,g\})}(X'_j) : h,g \in \mathbb{C}, \sum_{j=m(\hat{i}-1)+1}^{m\hat{i}} \mathbb{1}_{\mathrm{DIS}(\{h,g\})}(X'_j) = 0\right\}$$
$$\leq \max\left\{\sum_{j=m\lceil\log_2(2/\delta)\rceil+1}^{m\lceil\log_2(2/\delta)\rceil+\ell} \mathbb{1}_{\mathrm{DIS}(\{h,g\})}(X'_j) : h,g \in \mathbb{C}, \sum_{j=m(i^*-1)+1}^{mi^*} \mathbb{1}_{\mathrm{DIS}(\{h,g\})}(X'_j) = 0\right\} \leq \frac{3}{4}\varepsilon\ell,$$

where the first inequality is by definition of $\hat{i}$, and the second inequality is by a combination of (8) with (7). Therefore, by (6), on the event $E' \cap E$, we have

$$\max\left\{\mathcal{P}(\mathrm{DIS}(\{h,g\})) : h,g \in \mathbb{C}, \sum_{j=m(\hat{i}-1)+1}^{m\hat{i}} \mathbb{1}_{\mathrm{DIS}(\{h,g\})}(X'_j) = 0\right\} \leq \varepsilon,$$

or equivalently, $N_{\hat{i}}$ is an ε-net of $\mathcal{P}$ for $\{\mathrm{DIS}(\{h,g\}) : h,g \in \mathbb{C}\}$. To complete the proof, we take $c' = \max\{10c_0, 6c\}$, and note that the event $E' \cap E$ has probability at least $1-\delta$ by a union bound. ∎

There are also variants of the above two lemmas applicable to sample compression schemes. Specifically, the next lemma is due to Littlestone and Warmuth (1986); Floyd and Warmuth (1995).

Lemma 23 *There exists a finite universal constant $\tilde{c} \geq 1$ such that, for any collection $\mathcal{T}$ of measurable subsets of $\mathcal{X}$, any $n \in \mathbb{N} \cup \{0\}$, and any function $\phi_n : \mathcal{X}^n \to \mathcal{T}$, for any $\varepsilon, \delta \in (0,1)$, for any integer $m \geq \frac{\tilde{c}}{\varepsilon}\left(n\mathrm{Log}\left(\frac{1}{\varepsilon}\right) + \mathrm{Log}\left(\frac{1}{\delta}\right)\right)$, for any probability measure $\mathcal{P}$ over $\mathcal{X}$, if $X'_1, \ldots, X'_m$ are independent $\mathcal{P}$-distributed random variables, then with probability at least $1-\delta$, it holds that every $i_1, \ldots, i_n \in \{1, \ldots, m\}$ with $i_1 \leq \cdots \leq i_n$ and $\{X'_1, \ldots, X'_m\} \cap \phi_n(X'_{i_1}, \ldots, X'_{i_n}) = \emptyset$ has $\mathcal{P}\left(\phi_n(X'_{i_1}, \ldots, X'_{i_n})\right) \leq \varepsilon$: that is, $\{X'_1, \ldots, X'_m\}$ is an ε-net of $\mathcal{P}$ for $\{\phi_n(X'_{i_1}, \ldots, X'_{i_n}) : i_1, \ldots, i_n \in \{1, \ldots, m\}, i_1 \leq \cdots \leq i_n\}$.*

This implies the following result.

Lemma 24 *There exists a finite universal constant $\tilde{c}' \geq 1$ such that, for any collection $\mathcal{T}$ of measurable subsets of $\mathcal{X}$, any $n \in \mathbb{N}$, and any function $\phi_n : \mathcal{X}^n \times \mathcal{Y}^n \to \mathcal{T}$, for any probability measure $\mathcal{P}$ on $\mathcal{X}$, if $X'_1, X'_2, \ldots$ are independent $\mathcal{P}$-distributed random variables, then for any $\varepsilon, \delta \in (0,1)$, for any integers $m \geq \frac{\tilde{c}'n}{\varepsilon}\mathrm{Log}\left(\frac{1}{\varepsilon}\right)$ and $\ell \geq \frac{\tilde{c}'}{\varepsilon}\left(n\mathrm{Log}\left(\frac{m}{n}\right) + \mathrm{Log}\left(\frac{1}{\delta}\right)\right)$, defining $N_i = \{X'_{m(i-1)+1}, \ldots, X'_{mi}\}$ for each $i \in \{1, \ldots, \lceil\log_2(2/\delta)\rceil\}$, letting*

$$\hat{i} = \operatorname*{argmin}_{i \in \{1,\ldots,\lceil\log_2(2/\delta)\rceil\}} \max\left\{\sum_{j=m\lceil\log_2(2/\delta)\rceil+1}^{m\lceil\log_2(2/\delta)\rceil+\ell} \mathbb{1}_{\phi_n(X'_{i_1},\ldots,X'_{i_n},y_1,\ldots,y_n)}(X'_j) : y_1, \ldots, y_n \in \mathcal{Y},\right.$$
$$\left. m(i-1) < i_1 \leq \cdots \leq i_n \leq mi, \sum_{j=m(i-1)+1}^{mi} \mathbb{1}_{\phi_n(X'_{i_1},\ldots,X'_{i_n},y_1,\ldots,y_n)}(X'_j) = 0\right\} \cup \{0\},$$

and $\hat{N}=N_{\hat{i}}$*, with probability at least* $1-\delta$*,* $\hat{N}$ *is an* ε*-net of* $\mathcal{P}$ *for* $\{\phi_n(X'_{i_1},\ldots,X'_{i_n},y_1,\ldots,y_n) : m(\hat{i}-1)<i_1\leq\cdots\leq i_n\leq m\hat{i}, y_1,\ldots,y_n\in\mathcal{Y}\}$.

Proof Let $\tilde{c}$ be as in Lemma 23, define $\tilde{c}'=\max\{8\tilde{c},128\}$, and let m and ℓ be as described in the lemma statement. Noting that $\frac{2\tilde{c}}{\varepsilon}\left(n\text{Log}\left(\frac{2}{\varepsilon}\right)+\text{Log}\left(2^{n+1}\right)\right)\leq\frac{8\tilde{c}n}{\varepsilon}\text{Log}\left(\frac{1}{\varepsilon}\right)$, we have that $m\geq\frac{2\tilde{c}}{\varepsilon}\left(n\text{Log}\left(\frac{2}{\varepsilon}\right)+\text{Log}\left(2^{n+1}\right)\right)$. Thus, by Lemma 23, for each $i\in\{1,\ldots,\lceil\log_2(2/\delta)\rceil\}$ and $y_1,\ldots,y_n\in\mathcal{Y}$, with probability at least $1-2^{-n-1}$, $\left\{X'_{m(i-1)+1},\ldots,X'_{mi}\right\}$ is an $\frac{\varepsilon}{2}$-net of $\mathcal{P}$ for $\left\{\phi_n(X'_{i_1},\ldots,X'_{i_n},y_1,\ldots,y_n) : m(i-1)<i_1\leq\cdots\leq i_n\leq mi\right\}$. By a union bound, this holds simultaneously for all $y_1,\ldots,y_n\in\mathcal{Y}$ with probability at least $\frac{1}{2}$. In particular, since the $\left\{X'_{m(i-1)+1},\ldots,X'_{mi}\right\}$ subsequences are independent over values of i, we have that there is an event E of probability at least $1-\left(\frac{1}{2}\right)^{\lceil\log_2(2/\delta)\rceil}\geq 1-\frac{\delta}{2}$, on which $\exists i^*\in\{1,\ldots,\lceil\log_2(2/\delta)\rceil\}$ such that $\left\{X'_{m(i^*-1)+1},\ldots,X'_{mi^*}\right\}$ is an $\frac{\varepsilon}{2}$-net of $\mathcal{P}$ for $\left\{\phi_n(X'_{i_1},\ldots,X'_{i_n},y_1,\ldots,y_n) : m(i^*-1)<i_1\leq\cdots\leq i_n\leq mi^*, y_1,\ldots,y_n\in\mathcal{Y}\right\}$.

For any $i\in\{1,\ldots,\lceil\log_2(2/\delta)\rceil\}$, any $i_1,\ldots,i_n\in\{m(i-1)+1,\ldots,mi\}$ with $i_1\leq\cdots\leq i_n$, and any $y_1,\ldots,y_n\in\mathcal{Y}$, Chernoff bounds (applied under the conditional distribution given $X'_{i_1},\ldots,X'_{i_n}$) and the law of total probability imply that, with probability at least $1-\exp\{-\varepsilon\ell/32\}$, if $\mathcal{P}\left(\phi_n(X'_{i_1},\ldots,X'_{i_n},y_1,\ldots,y_n)\right)\leq\frac{\varepsilon}{2}$, then

$$\sum_{j=m\lceil\log_2(2/\delta)\rceil+1}^{m\lceil\log_2(2/\delta)\rceil+\ell}\mathbb{1}_{\phi_n(X'_{i_1},\ldots,X'_{i_n},y_1,\ldots,y_n)}(X'_j)\leq\frac{3}{4}\varepsilon\ell,$$

while if $\mathcal{P}\left(\phi_n(X'_{i_1},\ldots,X'_{i_n},y_1,\ldots,y_n)\right)>\varepsilon$, then

$$\sum_{j=m\lceil\log_2(2/\delta)\rceil+1}^{m\lceil\log_2(2/\delta)\rceil+\ell}\mathbb{1}_{\phi_n(X'_{i_1},\ldots,X'_{i_n},y_1,\ldots,y_n)}(X'_j)>\frac{3}{4}\varepsilon\ell.$$

The number of distinct nondecreasing sequences $(i_1,\ldots,i_n)\in\{m(i-1)+1,\ldots,mi\}^n$ is $\binom{n+m-1}{n}\leq\left(\frac{2em}{n}\right)^n$. Therefore, by a union bound, there exists an event E' of probability at least

$$1-2^n\left(\frac{2em}{n}\right)^n\lceil\log_2(2/\delta)\rceil\exp\{-\varepsilon\ell/32\},$$

on which this holds for every such $y_1,\ldots,y_n,i,i_1,\ldots,i_n$. Noting that

$$\frac{32}{\varepsilon}\text{Log}\left(2^n\lceil\log_2(2/\delta)\rceil\left(\frac{2em}{n}\right)^n\frac{2}{\delta}\right)\leq\frac{128}{\varepsilon}\left(n\text{Log}\left(\frac{m}{n}\right)+\text{Log}\left(\frac{1}{\delta}\right)\right)\leq\ell,$$

we have that E' has probability at least $1-\frac{\delta}{2}$.

In particular, defining for each $i\in\{1,\ldots,\lceil\log_2(2/\delta)\rceil\}$,

$$\hat{p}_i=\max\left\{\sum_{j=m\lceil\log_2(2/\delta)\rceil+1}^{m\lceil\log_2(2/\delta)\rceil+\ell}\mathbb{1}_{\phi_n(X'_{i_1},\ldots,X'_{i_n},y_1,\ldots,y_n)}(X'_j) : y_1,\ldots,y_n\in\mathcal{Y},\right.$$
$$\left.m(i-1)<i_1\leq\cdots\leq i_n\leq mi,\sum_{j=m(i-1)+1}^{mi}\mathbb{1}_{\phi_n(X'_{i_1},\ldots,X'_{i_n},y_1,\ldots,y_n)}(X'_j)=0\right\}\cup\{0\},$$

we have that, on $E \cap E'$, $\hat{p}_{i^*} \leq \frac{3}{4}\varepsilon\ell$. Furthermore, for every $i \in \{1, \ldots, \lceil \log_2(2/\delta) \rceil\}$ for which $\left\{X'_{m(i-1)+1}, \ldots, X'_{mi}\right\}$ is *not* an ε-net of $\mathcal{P}$ for $\left\{\phi_n(X'_{i_1}, \ldots, X'_{i_n}, y_1, \ldots, y_n) : m(i-1) < i_1 \leq \cdots \leq i_n \leq mi, y_1, \ldots, y_n \in \mathcal{Y}\right\}$, by definition $\exists i_1, \ldots, i_n \in \{m(i-1)+1, \ldots, mi\}$ with $i_1 \leq \cdots \leq i_n$, and $y_1, \ldots, y_n \in \mathcal{Y}$, such that $\mathcal{P}\left(\phi_n(X'_{i_1}, \ldots, X'_{i_n}, y_1, \ldots, y_n)\right) > \varepsilon$ while $\sum_{j=m(i-1)+1}^{mi} \mathbb{1}_{\phi_n(X'_{i_1}, \ldots, X'_{i_n}, y_1, \ldots, y_n)}(X'_j) = 0$; thus, on the event E',

$$\sum_{j=m\lceil \log_2(2/\delta) \rceil + 1}^{m\lceil \log_2(2/\delta) \rceil + \ell} \mathbb{1}_{\phi_n(X'_{i_1}, \ldots, X'_{i_n}, y_1, \ldots, y_n)}(X'_j) > \frac{3}{4}\varepsilon\ell$$

for this choice of $i_1, \ldots, i_n, y_1, \ldots, y_n$. In particular, this implies that $\hat{p}_i > \frac{3}{4}\varepsilon\ell$. Altogether, we have that on the event $E \cap E'$, any such i has $\hat{p}_{\hat{i}} \leq \hat{p}_{i^*} \leq \frac{3}{4}\varepsilon\ell < \hat{p}_i$, so that $\hat{i} \neq i$. Therefore, on the event $E \cap E'$, $\left\{X'_{m(\hat{i}-1)+1}, \ldots, X'_{m\hat{i}}\right\}$ is an ε-net of $\mathcal{P}$ for $\left\{\phi_n(X'_{i_1}, \ldots, X'_{i_n}, y_1, \ldots, y_n) : m(\hat{i}-1) < i_1 \leq \cdots \leq i_n \leq m\hat{i}, y_1, \ldots, y_n \in \mathcal{Y}\right\}$.

To complete the proof, we note that the event $E \cap E'$ has probability at least $1 - \delta$ by a union bound. ∎

A.2 Lower Bound Constructions for Noisy Settings

Fix any $\zeta \in (0, 1]$, $\beta \in [0, 1/2)$, and $k \in \mathbb{N}$ with $k \leq 1/\zeta$. Let $\mathcal{X}_k = \{x_1, \ldots, x_{k+1}\}$ be any $k+1$ distinct elements of $\mathcal{X}$ (assuming $|\mathcal{X}| \geq k+1$), and let $\mathbb{C}_k = \{x \mapsto 2\mathbb{1}_{\{x_i\}}(x) - 1 : i \in \{1, \ldots, k\}\}$, a set of functions mapping $\mathcal{X}$ to $\{-1, +1\}$. Let $\mathcal{P}_{k,\zeta}$ be a probability measure over $\mathcal{X}$ with $\mathcal{P}_{k,\zeta}(\{x_i\}) = \zeta$ for each $i \in \{1, \ldots, k\}$, and $\mathcal{P}_{k,\zeta}(\{x_{k+1}\}) = 1 - \zeta k$. For each $t \in \{1, \ldots, k\}$, let $P'_{k,\zeta,t}$ denote the probability measure over $\mathcal{X} \times \mathcal{Y}$ having marginal distribution $\mathcal{P}_{k,\zeta}$ over $\mathcal{X}$, such that if $(X, Y) \sim P'_{k,\zeta,t}$, then every $i \in \{1, \ldots, k\}$ has $\mathbb{P}(Y = 2\mathbb{1}_{\{x_t\}}(X) - 1 | X = x_i) = 1 - \beta$, and furthermore $\mathbb{P}(Y = -1 | X = x_{k+1}) = 1$. Finally, define $\mathrm{RR}'(k, \zeta, \beta) = \left\{P'_{k,\zeta,t} : t \in \{1, \ldots, k\}\right\}$. Raginsky and Rakhlin (2011) prove the following result (see the proof of their Theorem 2).[13]

Lemma 25 *For ζ, β, k as above, if $k \geq 2$ and $\mathbb{C}_k \subseteq \mathbb{C}$, then for any $\delta \in (0, 1/4)$,*

$$\Lambda_{\mathrm{RR}'(k,\zeta,\beta)}((\zeta/2)(1-2\beta), \delta) \geq \frac{\beta k \ln\left(\frac{1}{4\delta}\right)}{3(1-2\beta)^2}.$$

This has the following immediate implication for general $\mathcal{X}$ and $\mathbb{C}$. Fix any $\zeta \in (0, 1]$ and $\beta \in [0, 1/2)$, let $k \in \mathbb{N} \cup \{0\}$ satisfy $k \leq \min\{\mathfrak{s} - 1, \lfloor 1/\zeta \rfloor\}$, and let $x_1, \ldots, x_{k+1}$ and $h_0, h_1, \ldots, h_k$ be as in Definition 2. Let $\mathcal{P}_{k,\zeta}$ be as above (for this choice of $x_1, \ldots, x_{k+1}$), and for each $t \in \{1, \ldots, k\}$, let $P_{k,\zeta,t}$ denote the probability measure over $\mathcal{X} \times \mathcal{Y}$ having

13. Technically, the proof of Raginsky and Rakhlin (2011, Theorem 2) relies on a lemma (their Lemma 4), with various conditions on both k and a parameter "d" in their construction. However, one can easily verify that the conclusions of that lemma continue to hold (in fact, with improved constants) in our special case (corresponding to $d = 1$ and arbitrary $k \in \mathbb{N}$) by defining $\mathcal{M}_{k,1} = \{0, 1\}_1^k$ in their construction.

marginal distribution $\mathcal{P}_{k,\zeta}$ over $\mathcal{X}$, such that if $(X,Y) \sim P_{k,\zeta,t}$, then every $i \in \{1,\ldots,k\}$ has $\mathbb{P}(Y = h_t(X)|X = x_i) = 1-\beta$, and furthermore $\mathbb{P}(Y = h_t(X)|X = x_{k+1}) = 1$. Define $\mathrm{RR}(k,\zeta,\beta) = \{P_{k,\zeta,t} : t \in \{1,\ldots,k\}\}$. We have the following result.

Lemma 26 *For k, ζ, β as above, for any $\delta \in (0, 1/4)$,*

$$\Lambda_{\mathrm{RR}(k,\zeta,\beta)}((\zeta/2)(1-2\beta),\delta) \geq \frac{\beta(k-1)\ln\left(\frac{1}{4\delta}\right)}{3(1-2\beta)^2}.$$

Proof First note that if $k \leq 1$, then the lemma trivially holds (since $\Lambda_{\mathrm{RR}(k,\zeta,\beta)}(\cdot,\cdot) \geq 0$). For this same reason, the result also trivially holds if $\beta = 0$. Otherwise, suppose $k \geq 2$ and $\beta > 0$, and fix any n less than the right hand side of the above inequality. Let $\mathcal{A}$ be any active learning algorithm, and consider the following modification $\mathcal{A}'$ of $\mathcal{A}$. For any given sequence $X_1, X_2, \ldots$ of unlabeled data, $\mathcal{A}'(n)$ simulates the execution of $\mathcal{A}(n)$, except that when $\mathcal{A}(n)$ would request the label Y_i of a point X_i in the sequence, $\mathcal{A}'(n)$ requests the label Y_i, but proceeds as $\mathcal{A}(n)$ would if the label value had been $-Y_i h_0(X_i)$ instead of Y_i. When the simulation of $\mathcal{A}(n)$ concludes, if $\hat{h}$ is its return value, $\mathcal{A}'(n)$ instead returns the function $x \mapsto \hat{h}'(x) = -\hat{h}(x)h_0(x)$.

Now fix a probability measure $P'_{k,\zeta,t} \in \mathrm{RR}'(k,\zeta,\beta)$ minimizing the probability that $\mathrm{er}_{P'_{k,\zeta,t}}(\hat{h}') - \inf_{h\in\mathbb{C}_k} \mathrm{er}_{P'_{k,\zeta,t}}(h) \leq (\zeta/2)(1-2\beta)$ when $\mathcal{A}'$ is run with $\mathcal{P}_{XY} = P'_{k,\zeta,t}$, and let $(X,Y) \sim P'_{k,\zeta,t}$. Note that the marginal distribution of $P'_{k,\zeta,t}$ over $\mathcal{X}$ is $\mathcal{P}_{k,\zeta}$, that for any $i \in \{1,\ldots,k\}$, $\mathbb{P}(-Yh_0(X) = h_t(X)|X = x_i) = \mathbb{P}(Y = 2\mathbb{1}_{\{x_t\}}(X) - 1|X = x_i) = 1-\beta$, and that $\mathbb{P}(-Yh_0(X) = h_t(X)|X = x_{k+1}) = \mathbb{P}(Y = -1|X = x_{k+1}) = 1$. In particular, this implies $(X, -Yh_0(X)) \sim P_{k,\zeta,t}$. Therefore, running the active learning algorithm $\mathcal{A}'(n)$ with a sequence $(X_1,Y_1),(X_2,Y_2),\ldots$ of independent $P'_{k,\zeta,t}$-distributed samples, the algorithm behaves as $\mathcal{A}(n)$ would under $P_{k,\zeta,t}$, except that its returned classifier is $\hat{h}'$ instead of $\hat{h}$. Next, note that

$$\begin{aligned}
\mathrm{er}_{P'_{k,\zeta,t}}(\hat{h}') &= \mathbb{P}(-\hat{h}(X)h_0(X) \neq Y) \\
&= \mathbb{E}[\mathbb{P}(\hat{h}(X) \neq -Y|X)\mathbb{1}[h_0(X) = 1] + \mathbb{P}(\hat{h}(X) \neq Y|X)\mathbb{1}[h_0(X) = -1]] \\
&= \mathbb{P}(\hat{h}(X) \neq -Yh_0(X)) = \mathrm{er}_{P_{k,\zeta,t}}(\hat{h}),
\end{aligned}$$

and furthermore

$$\inf_{h\in\mathbb{C}_k} \mathrm{er}_{P'_{k,\zeta,t}}(h) = \mathrm{er}_{P'_{k,\zeta,t}}(2\mathbb{1}_{\{x_t\}} - 1) = \beta\zeta k = \mathrm{er}_{P_{k,\zeta,t}}(h_t) = \inf_{h\in\mathbb{C}} \mathrm{er}_{P_{k,\zeta,t}}(h).$$

Thus, if $\mathrm{er}_{P_{k,\zeta,t}}(\hat{h}) - \inf_{h\in\mathbb{C}} \mathrm{er}_{P_{k,\zeta,t}}(h) \leq (\zeta/2)(1-2\beta)$, then we must also have $\mathrm{er}_{P'_{k,\zeta,t}}(\hat{h}') - \inf_{h\in\mathbb{C}_k} \mathrm{er}_{P'_{k,\zeta,t}}(h) \leq (\zeta/2)(1-2\beta)$. Since $n < \frac{\beta k \ln\left(\frac{1}{4\delta}\right)}{3(1-2\beta)^2}$, Lemma 25 implies that (for this choice of $P'_{k,\zeta,t}$) $\mathcal{A}'(n)$ achieves the latter guarantee with probability strictly less than $1-\delta$, and therefore the corresponding $P_{k,\zeta,t} \in \mathrm{RR}(k,\zeta,\beta)$ is such that $\mathcal{A}(n)$ has probability strictly less than $1-\delta$ of achieving $\mathrm{er}_{P_{k,\zeta,t}}(\hat{h}) - \inf_{h\in\mathbb{C}} \mathrm{er}_{P_{k,\zeta,t}}(h) \leq (\zeta/2)(1-2\beta)$. Since this argument applies to any active learning algorithm $\mathcal{A}$, the result follows. ■

A.3 Finite Approximation of VC Classes

For a given probability measure $\mathcal{P}$ over $\mathcal{X}$, Adams and Nobel (2012) have proven that for any $\tau > 0$, if $d < \infty$, there exist disjoint measurable sets $A_1, \ldots, A_k$ (for some $k \in \mathbb{N}$) with $\bigcup_i A_i = \mathcal{X}$ such that, $\forall h \in \mathbb{C}$, $\mathcal{P}\left(\bigcup\{A_i : \exists x, y \in A_i \text{ s.t. } h(x) \neq h(y)\}\right) < \tau$: that is, every $h \in \mathbb{C}$ is constant on all of the sets A_i, except a few of them whose total probability is at most τ. This property has implications for bracketing behavior in VC classes, and was proven in the context of establishing uniform laws of large numbers for VC classes under stationary ergodic processes (see also Adams and Nobel, 2010; van Handel, 2013).

For our purposes, this result has the appealing feature that it allows one to effectively *discretize* the space $\mathcal{X}$ by partitioning it into subsets, with the guarantee that with high probability over the random choice of a point x, any other point y in the same cell in the partition as x will have $f^\star_{\mathcal{P}_{XY}}(x) = f^\star_{\mathcal{P}_{XY}}(y)$, for any $\mathcal{P}_{XY} \in \bigcup_{\nu \in [0,1/2)} \mathrm{BE}(\nu)$. However, before we can make use of this property, we must first address the fact that the construction of these sets A_i by Adams and Nobel (2012) requires a strong dependence on $\mathcal{P}$, to the extent that it is not obvious that this dependence can be supplanted by a data-dependent construction. However, it turns out that if we relax the requirement that the classifiers be *constant* in these cells, instead settling for being *nearly-constant*, then it is straightforward to construct a partition $A_1, \ldots, A_k$ satisfying the requirement. Specifically, we have the following result.

Lemma 27 *Fix any $\tau, \delta \in (0,1)$, and let $m_{\tau,\delta} = \left\lceil \frac{c}{\tau}\left(d\mathrm{Log}\left(\frac{1}{\tau}\right) + \mathrm{Log}\left(\frac{1}{\delta}\right)\right)\right\rceil$ (for c as in Lemma 21). For any probability measure $\mathcal{P}$ over $\mathcal{X}$, for any independent $\mathcal{P}$-distributed random variables $X'_1, \ldots, X'_{m_{\tau,\delta}}$, with probability at least $1-\delta$, letting $\mathbb{C}_{\tau,\delta} = \mathbb{C}[(X'_1, \ldots, X'_{m_{\tau,\delta}})]$ (as defined in Section 7.3), the collection of disjoint sets*

$$J_{\tau,\delta} = \left\{ \bigcap_{g \in \mathbb{C}[(X'_1,\ldots,X'_{m_{\tau,\delta}})]} \mathcal{X}_g : \forall g \in \mathbb{C}_{\tau,\delta}, \mathcal{X}_g \in \{\{x : g(x) = +1\}, \{x : g(x) = -1\}\} \right\}$$

is a partition of $\mathcal{X}$ with the property that, $\forall h \in \mathbb{C}$,

$$\sum_{A \in J_{\tau,\delta}} \min_{y \in \mathcal{Y}} \mathcal{P}(x \in A : h(x) = y) \leq \tau,$$

and $\forall \varepsilon > 0$, $\forall h \in \mathbb{C}$,

$$\mathcal{P}\left(\bigcup \left\{ A \in J_{\tau,\delta} : \min_{y \in \mathcal{Y}} \mathcal{P}(x \in A : h(x) = y) > \varepsilon \mathcal{P}(A) \right\}\right) \leq \frac{\tau}{\varepsilon}.$$

Proof By Lemma 21, with probability at least $1 - \delta$, $\mathbb{C}_{\tau,\delta}$ is a τ-cover of $\mathbb{C}$. Furthermore, note that for every $g \in \mathbb{C}_{\tau,\delta}$ and every $A \in J_{\tau,\delta}$, either every $x \in A$ has $g(x) = +1$ or every $x \in A$ has $g(x) = -1$ (i.e., g is constant on A). Therefore, $\forall h \in \mathbb{C}$,

$$\begin{aligned}\sum_{A \in J_{\tau,\delta}} \min_{y \in \mathcal{Y}} \mathcal{P}(x \in A : h(x) = y) &\leq \sum_{A \in J_{\tau,\delta}} \min_{g \in \mathbb{C}_{\tau,\delta}} \mathcal{P}(x \in A : h(x) \neq g(x)) \\ &\leq \min_{g \in \mathbb{C}_{\tau,\delta}} \sum_{A \in J_{\tau,\delta}} \mathcal{P}(x \in A : h(x) \neq g(x)) = \min_{g \in \mathbb{C}_{\tau,\delta}} \mathcal{P}(x : h(x) \neq g(x)) \leq \tau.\end{aligned}$$

The final claim follows by Markov's inequality, since on the above event, $\forall \varepsilon > 0$, $\forall h \in \mathbb{C}$,

$$\begin{aligned}
&\mathcal{P}\left(\bigcup\left\{A \in J_{\tau,\delta} : \min_{y\in\mathcal{Y}} \mathcal{P}(x \in A : h(x) = y) > \varepsilon \mathcal{P}(A)\right\}\right) \\
&= \mathcal{P}\left(\bigcup\left\{A \in J_{\tau,\delta} : \mathcal{P}(A) > 0, \min_{y\in\mathcal{Y}} \mathcal{P}(x \in A : h(x) = y) > \varepsilon \mathcal{P}(A)\right\}\right) \\
&= \mathcal{P}\left(\bigcup\left\{A \in J_{\tau,\delta} : \mathcal{P}(A) > 0, \min_{y\in\mathcal{Y}} \frac{\mathcal{P}(x \in A : h(x) = y)}{\mathcal{P}(A)} > \varepsilon\right\}\right) \\
&\leq \frac{1}{\varepsilon} \sum_{A\in J_{\tau,\delta}} \mathcal{P}(A) \min_{y\in\mathcal{Y}} \frac{\mathcal{P}(x \in A : h(x) = y)}{\mathcal{P}(A)} = \frac{1}{\varepsilon} \sum_{A\in J_{\tau,\delta}} \min_{y\in\mathcal{Y}} \mathcal{P}(x \in A : h(x) = y) \leq \frac{\tau}{\varepsilon}.
\end{aligned}$$

■

Appendix B. Proofs for Results in Section 5

This section provides proofs of the main results of this article.

B.1 The Realizable Case

We begin with the particularly-simple case of Theorem 3.

Proof of Theorem 3 The lower bounds proportional to d and $\mathrm{Log}\left(\min\left\{\frac{1}{\varepsilon}, |\mathbb{C}|\right\}\right)$ are due to Kulkarni, Mitter, and Tsitsiklis (1993) (lower bound in terms of the covering numbers) in conjunction with Kulkarni (1989); Kulkarni, Mitter, and Tsitsiklis (1993) (lower bounds on the worst-case covering numbers). Specifically, Kulkarni, Mitter, and Tsitsiklis (1993) study the problem of learning from arbitrary binary-valued queries. Since active learning receives binary responses in the binary classification setting, it is a special case of this type of algorithm. In particular, for any probability measure $\mathcal{P}$ over $\mathcal{X}$, and $\varepsilon \in (0,1)$, let $\mathcal{N}(\varepsilon, \mathbb{C}, \mathcal{P})$ denote the minimum cardinality $|\mathcal{H}|$ over all ε-covers $\mathcal{H}$ of $\mathbb{C}$ (under the $\mathcal{P}(\mathrm{DIS}(\{\cdot,\cdot\}))$ pseudometric), or else $\mathcal{N}(\varepsilon, \mathbb{C}, \mathcal{P}) = \infty$ if no finite ε-cover of $\mathbb{C}$ exists. Then the lower bound of Kulkarni, Mitter, and Tsitsiklis (1993, Theorem 3) implies that, $\forall \varepsilon, \delta \in (0, 1/2)$,

$$\Lambda_{\mathrm{RE}}(\varepsilon, \delta) \geq \sup_{\mathcal{P}} \left\lceil \log_2 \left((1-\delta)\mathcal{N}(2\varepsilon, \mathbb{C}, \mathcal{P})\right)\right\rceil. \tag{9}$$

Furthermore, the construction in the proof of Kulkarni, Mitter, and Tsitsiklis (1993, Lemma 2) implies that $\sup_{\mathcal{P}} \mathcal{N}(2\varepsilon, \mathbb{C}, \mathcal{P}) \geq \min\left\{\left\lfloor \frac{1}{4\varepsilon}\right\rfloor, |\mathbb{C}|\right\}$, so that combined with (9), we have

$$\Lambda_{\mathrm{RE}}(\varepsilon, \delta) \geq \left\lceil \log_2 \left((1-\delta)\min\left\{\left\lfloor \frac{1}{4\varepsilon}\right\rfloor, |\mathbb{C}|\right\}\right)\right\rceil.$$

For $\delta \in (0, 1/3)$ and $\varepsilon \in (0, 1/8)$, and since $|\mathbb{C}| \geq 3$ (by assumption, intended to focus on non-trivial cases to simplify the expressions), the right hand side is at least $\frac{1}{4}\mathrm{Log}\left(\min\left\{\frac{1}{\varepsilon}, |\mathbb{C}|\right\}\right)$. Furthermore, if $d < 162$, this already implies that for any $\varepsilon \in (0, 1/3)$ and $\delta \in (0, 1/3)$, $\Lambda_{\mathrm{RE}}(\varepsilon, \delta) \geq \frac{1}{4}\ln(3) \geq \frac{d}{648}$. Otherwise, in the case that $d \geq 162$, Kulkarni (1989, Proposition 3) proves that, if $\varepsilon \in (0, 1/9)$, $\sup_{\mathcal{P}} \mathcal{N}(2\varepsilon, \mathbb{C}, \mathcal{P}) \geq \exp\left\{2\left(\frac{1}{2} - 4\varepsilon\right)^2 d\right\} \geq \exp\{d/162\}$.

Combined with (9), this implies that for $\varepsilon \in (0,1/9)$ and $\delta \in (0,1/3)$, if $d \geq 162$, then

$$\Lambda_{\mathrm{RE}}(\varepsilon,\delta) \geq \left\lceil \log_2\left(\frac{2}{3}e^{d/162}\right)\right\rceil \geq \frac{d}{162}\log_2(e) - \log_2\left(\frac{3}{2}\right) \geq \frac{d}{162}\log_2\left(\frac{2e}{3}\right) \geq \frac{d}{189}.$$

Thus, regardless of the value of d, we have $\Lambda_{\mathrm{RE}}(\varepsilon,\delta) \geq \frac{d}{648}$.

For the final part of the proof of the lower bound, a lower bound proportional to $\mathfrak{s} \wedge \frac{1}{\varepsilon}$ may be credited to Dasgupta (2005, 2004). It can be proven as follows. Let $x_1,\ldots,x_{\mathfrak{s}}$ and $h_0,h_1,\ldots,h_{\mathfrak{s}}$ be as in Definition 2, let $t = \mathfrak{s} \wedge \left\lceil \frac{1-\varepsilon}{\varepsilon}\right\rceil$, and let us restrict the discussion to those $t+1$ distributions $\mathcal{P}_{XY} \in \mathrm{RE}$ such that the marginal distribution $\mathcal{P}$ of $\mathcal{P}_{XY}$ over $\mathcal{X}$ is uniform on $\{x_1,\ldots,x_t\}$, and $f^\star_{\mathcal{P}_{XY}} \in \{h_0,h_1,\ldots,h_t\}$. Then for any active learning algorithm $\mathcal{A}$, for any $n \leq t/2$, let Q_i denote the (possibly random) set of (at most n) points X_i that $\mathcal{A}(n)$ requests the labels of, given that $f^\star_{\mathcal{P}_{XY}} = h_i$ (for $i \in \{0,\ldots,t\}$), and let $\hat{h}_i$ denote the classifier returned by $\mathcal{A}(n)$ in this case. Since the marginal distribution of $\mathcal{P}_{XY}$ over $\mathcal{X}$ is fixed to $\mathcal{P}$ for all $t+1$ of these $\mathcal{P}_{XY}$ distributions, we may consider the sequence $X_1,X_2,\ldots$ of i.i.d. $\mathcal{P}$-distributed random variables to be identical over these $t+1$ possible choices of $\mathcal{P}_{XY}$, without affecting the distributions of Q_i and $\hat{h}_i$ (see Kallenberg, 2002). Thus, we may note that $\hat{h}_i = \hat{h}_0$ whenever $x_i \notin Q_0$, since $x_i \notin Q_0$ implies that all of the labels observed by the algorithm are identical to those that would be observed if $f^\star_{\mathcal{P}_{XY}} = h_0$ instead of $f^\star_{\mathcal{P}_{XY}} = h_i$. Now, if it holds that $\mathbb{P}\left(\mathcal{P}\left(x : \hat{h}_0(x) \neq h_0(x)\right) > \varepsilon\right) \leq \delta$, then since every x_i with $i \leq t$ has $\mathcal{P}(\{x_i\}) > \varepsilon$, we have that $\mathbb{P}\left(\forall i \in \{1,\ldots,t\}, \hat{h}_0(x_i) = h_0(x_i)\right) \geq 1-\delta$. But if this holds, then it must also be true that

$$\begin{aligned}
&\max_{i\in\{1,\ldots,t\}} \mathbb{P}\left(\mathcal{P}(x : \hat{h}_i(x) \neq h_i(x)) > \varepsilon\right) \geq \frac{1}{t}\sum_{i=1}^{t}\mathbb{P}\left(\mathcal{P}(x : \hat{h}_i(x) \neq h_i(x)) > \varepsilon\right)\\
&\geq \frac{1}{t}\sum_{i=1}^{t}\mathbb{P}\left(\hat{h}_i(x_i) = h_0(x_i)\right) = \frac{1}{t}\mathbb{E}\left[\sum_{i=1}^{t}\mathbb{1}\left[\hat{h}_i(x_i) = h_0(x_i)\right]\right]\\
&\geq \frac{1}{t}\mathbb{E}\left[\sum_{i=1}^{t}\mathbb{1}\left[x_i \notin Q_0\right]\mathbb{1}\left[\hat{h}_i(x_i) = h_0(x_i)\right]\right] = \frac{1}{t}\mathbb{E}\left[\sum_{i=1}^{t}\mathbb{1}\left[x_i \notin Q_0\right]\mathbb{1}\left[\hat{h}_0(x_i) = h_0(x_i)\right]\right]\\
&\geq \frac{1}{t}\mathbb{E}\left[\mathbb{1}\left[\forall i \in \{1,\ldots,t\}, \hat{h}_0(x_i) = h_0(x_i)\right]\sum_{i=1}^{t}\mathbb{1}\left[x_i \notin Q_0\right]\right]\\
&\geq \frac{1}{t}\mathbb{E}\left[\mathbb{1}\left[\forall i \in \{1,\ldots,t\}, \hat{h}_0(x_i) = h_0(x_i)\right](t-n)\right]\\
&= \frac{t-n}{t}\mathbb{P}\left(\forall i \in \{1,\ldots,t\}, \hat{h}_0(x_i) = h_0(x_i)\right) \geq \frac{t-n}{t}(1-\delta) \geq \frac{1-\delta}{2} \geq \frac{1}{3} > \delta.
\end{aligned}$$

Thus, when $n \leq t/2$, at least one of these $t+1$ distributions $\mathcal{P}_{XY}$ (all of which are in RE) has $\mathbb{P}\left(\mathrm{er}_{\mathcal{P}_{XY}}(\mathcal{A}(n)) > \varepsilon\right) > \delta$. Since this argument holds for any $\mathcal{A}$, we have that $\Lambda_{\mathrm{RE}}(\varepsilon,\delta) > t/2 = \frac{1}{2}\min\left\{\mathfrak{s}, \left\lceil\frac{1-\varepsilon}{\varepsilon}\right\rceil\right\} \geq \frac{4}{9}\min\left\{\mathfrak{s}, \frac{1}{\varepsilon}\right\}$. Combined with the lower bounds proportional d and $\mathrm{Log}\left(\min\left\{\frac{1}{\varepsilon}, |\mathbb{C}|\right\}\right)$ established above, this completes the proof of the lower bound in Theorem 3.

The proof of the upper bound is in three parts. The first part, establishing the $\frac{d}{\varepsilon}\mathrm{Log}\left(\frac{1}{\varepsilon}\right)$ upper bound, is a straightforward application of Lemma 22. The second part, establishing

the $\frac{\mathfrak{s}d}{\mathrm{Log}(\mathfrak{s})}\mathrm{Log}\left(\frac{1}{\varepsilon}\right)$ upper bound, is directly based on techniques of Hanneke (2007a); Hegedüs (1995). Finally, and most involved, is the third part, establishing the $\mathfrak{s}\mathrm{Log}\left(\frac{1}{\varepsilon}\right)$ upper bound. This part is partly based on a recent technique of Wiener, Hanneke, and El-Yaniv (2015) for analyzing disagreement-based active learning (which refines an earlier technique of El-Yaniv and Wiener, 2010, 2012). Here, we modify this technique by using an ε-net in place of random samples, thereby refining logarithmic factors, and entirely eliminating the dependence on δ in the label complexity.

Fix any $\varepsilon, \delta \in (0,1)$. We begin with the $\frac{d}{\varepsilon}\mathrm{Log}\left(\frac{1}{\varepsilon}\right)$ upper bound. Let $m = \left\lceil \frac{c'd}{\varepsilon}\mathrm{Log}\left(\frac{1}{\varepsilon}\right)\right\rceil$ and $\ell = \left\lceil \frac{c'}{\varepsilon}\left(d\mathrm{Log}\left(\frac{1}{\varepsilon}\right) + \mathrm{Log}\left(\frac{1}{\delta}\right)\right)\right\rceil$, for c' as in Lemma 22. Define

$$\hat{i} = \operatorname*{argmin}_{i\in\{1,\ldots,\lceil\log_2(2/\delta)\rceil\}} \max_{h,g\in\mathbb{C}:\ \sum_{j=m(i-1)+1}^{mi}\mathbb{1}_{\mathrm{DIS}(\{h,g\})}(X_j)=0} \sum_{j=m\lceil\log_2(2/\delta)\rceil+1}^{m\lceil\log_2(2/\delta)\rceil+\ell} \mathbb{1}_{\mathrm{DIS}(\{h,g\})}(X_j).$$

Consider an active learning algorithm which, given a budget $n \in \mathbb{N}$, requests the labels Y_t for $t \in \left\{m\left(\hat{i}-1\right)+1, \ldots, m\left(\hat{i}-1\right)+\min\{m,n\}\right\}$, and returns any classifier $\hat{h}_n \in \mathbb{C}$ with $\sum_{t=m(\hat{i}-1)+1}^{m(\hat{i}-1)+\min\{m,n\}} \mathbb{1}\left[\hat{h}_n(X_t) \neq Y_t\right] = 0$ if such a classifier exists (and otherwise returns an arbitrary classifier). Note that, for $\mathcal{P}_{XY} \in \mathrm{RE}$, $\sum_{t=m(\hat{i}-1)+1}^{m(\hat{i}-1)+\min\{m,n\}} \mathbb{1}\left[f^\star_{\mathcal{P}_{XY}}(X_t) \neq Y_t\right] = 0$ with probability one, and since $f^\star_{\mathcal{P}_{XY}} \in \mathbb{C}$, $\hat{h}_n$ will have $\sum_{t=m(\hat{i}-1)+1}^{m(\hat{i}-1)+\min\{m,n\}} \mathbb{1}\left[\hat{h}_n(X_t) \neq Y_t\right] = 0$ with probability one. Furthermore, this implies $\sum_{t=m(\hat{i}-1)+1}^{m(\hat{i}-1)+\min\{m,n\}} \mathbb{1}\left[\hat{h}_n(X_t) \neq f^\star_{\mathcal{P}_{XY}}(X_t)\right] = 0$ with probability one. Additionally, Lemma 22 implies that, with probability at least $1-\delta$, the set $\left\{X_t : t \in \left\{m\left(\hat{i}-1\right)+1, \ldots, m\hat{i}\right\}\right\}$ is an ε-net of $\mathcal{P}$ for $\{\mathrm{DIS}(\{h,g\}) : h, g \in \mathbb{C}\}$. Since both $\hat{h}_n, f^\star_{\mathcal{P}_{XY}} \in \mathbb{C}$, this implies that if $n \geq m$, then with probability at least $1-\delta$, $\mathcal{P}\left(\mathrm{DIS}\left(\left\{\hat{h}_n, f^\star_{\mathcal{P}_{XY}}\right\}\right)\right) \leq \varepsilon$. Since $\mathcal{P}_{XY} \in \mathrm{RE}$, $\mathrm{er}_{\mathcal{P}_{XY}}\left(\hat{h}_n\right) = \mathcal{P}\left(\mathrm{DIS}\left(\left\{\hat{h}_n, f^\star_{\mathcal{P}_{XY}}\right\}\right)\right)$. Thus, if $n \geq m$, then with probability at least $1-\delta$, $\mathrm{er}_{\mathcal{P}_{XY}}\left(\hat{h}_n\right) \leq \varepsilon$. Since this holds for any $\mathcal{P}_{XY} \in \mathrm{RE}$, we have established that $\Lambda_{\mathrm{RE}}(\varepsilon,\delta) \leq m \leq \frac{2c'd}{\varepsilon}\mathrm{Log}\left(\frac{1}{\varepsilon}\right)$. This also completes the proof of the entire upper bound in Theorem 3 in the case $\mathfrak{s} = \infty$; for this reason, for the remainder of the proof below, we restrict our attention to the case $\mathfrak{s} < \infty$.

Next, we turn to proving the $\frac{\mathfrak{s}d}{\mathrm{Log}(\mathfrak{s})}\mathrm{Log}\left(\frac{1}{\varepsilon}\right)$ upper bound, based on a technique of Hanneke (2007a); Hegedüs (1995) (see also Hellerstein, Pillaipakkamnatt, Raghavan, and Wilkins, 1996 for related ideas), except using an ε-net in place of the random samples used by Hanneke (2007a). Let m and $\hat{i}$ be as above, and denote $\mathcal{U} = \left\{X_t : t \in \left\{m\left(\hat{i}-1\right)+1, \ldots, m\hat{i}\right\}\right\}$. The technique is based on using a general algorithm for *Exact* learning with membership queries, treating $\mathcal{U}$ as the instance space, and $\mathbb{C}[\mathcal{U}]$ as the concept space (where $\mathbb{C}[\mathcal{U}]$ is as defined in Section 7.3). Specifically, for any finite set $V \subseteq \mathbb{C}$ and any $x \in \mathcal{X}$, let $h_{\mathrm{maj}(V)}(x) = \mathrm{argmax}_{y\in\mathcal{Y}} |\{h \in V : h(x) = y\}|$ (breaking ties arbitrarily); $h_{\mathrm{maj}(V)}$ is called the *majority vote classifier*. In this context, the following algorithm is due to Hegedüs (1995) (see Section 7.3 for the definition of "specifying set").

MEMB-HALVING-2
Input: label budget n
Output: classifier $\hat{h}_n$

---

0. $V \leftarrow \mathbb{C}[\mathcal{U}]$, $t \leftarrow 0$
1. While $|V| \geq 2$ and $t < n$
2. $\quad \hat{h} \leftarrow h_{\mathrm{maj}(V)}$
3. $\quad$ Let $k = \mathrm{TD}(\hat{h}, \mathbb{C}[\mathcal{U}], \mathcal{U})$
4. $\quad$ Let $\{X_{j_1}, \ldots, X_{j_k}\} \in \mathcal{U}^k$ be a minimal specifying set for $\hat{h}$ on $\mathcal{U}$ with respect to $\mathbb{C}[\mathcal{U}]$
5. $\quad$ Repeat
6. $\quad\quad$ Let $\hat{j} = \operatorname*{argmin}_{j \in \{j_1, \ldots, j_k\}} |\{g \in V : g(X_j) = \hat{h}(X_j)\}|$
7. $\quad\quad$ Request the label $Y_{\hat{j}}$, let $t \leftarrow t + 1$
8. $\quad\quad$ $V \leftarrow \{h \in V : h(X_{\hat{j}}) = Y_{\hat{j}}\}$
9. $\quad$ Until $\hat{h}(X_{\hat{j}}) \neq Y_{\hat{j}}$ or $|V| \leq 1$ or $t = n$
10. Return any $\hat{h}_n$ in V (or $\hat{h}_n$ arbitrary if $V = \emptyset$)

Fix any $\mathcal{P}_{XY} \in \mathrm{RE}$, and note that we have $f^\star_{\mathcal{P}_{XY}} \in \mathbb{C}$, so that $\exists h^* \in \mathbb{C}[\mathcal{U}]$ with $h^*(x) = f^\star_{\mathcal{P}_{XY}}(x)$, $\forall x \in \mathcal{U}$. Since $Y_j = f^\star_{\mathcal{P}_{XY}}(X_j)$ for every j with probability one in this case, we have that with probability one the set V will be nonempty in Step 10, so that $\hat{h}_n$ is chosen from V; in particular, we have $h^*(X_j) = Y_j$ for every $X_j \in \mathcal{U}$, and hence $h^* \in V$ in Step 10. Furthermore, when this is the case, Hegedüs (1995) proves that, letting $\mathrm{XTD}(\mathbb{C}[\mathcal{U}], \mathcal{U}) = \max_{h: \mathcal{X} \to \mathcal{Y}} \mathrm{TD}(h, \mathbb{C}[\mathcal{U}], \mathcal{U})$ (see Section 7.3), if

$$n \geq 2 \frac{\mathrm{XTD}(\mathbb{C}[\mathcal{U}], \mathcal{U})}{1 \vee \log_2(\mathrm{XTD}(\mathbb{C}[\mathcal{U}], \mathcal{U}))} \log_2(|\mathbb{C}[\mathcal{U}]|),$$

then the classifier $\hat{h}_n$ returned by MEMB-HALVING-2 satisfies $\hat{h}_n = h^*$, so that $\hat{h}_n(x) = f^\star_{\mathcal{P}_{XY}}(x)$ for every $x \in \mathcal{U}$.[14] Since $\mathrm{XTD}(\mathbb{C}[\mathcal{U}], \mathcal{U}) \leq \mathrm{XTD}(\mathbb{C}, m)$, and Theorem 13 implies $\mathrm{XTD}(\mathbb{C}, m) = \mathfrak{s} \wedge m \leq \mathfrak{s}$, and since $\mathrm{Log}(\mathrm{XTD}(\mathbb{C}[\mathcal{U}], \mathcal{U})) \leq 1 \vee \log_2(\mathrm{XTD}(\mathbb{C}[\mathcal{U}], \mathcal{U}))$ and $x \mapsto \frac{x}{\mathrm{Log}(x)}$ is nondecreasing on $\mathbb{N} \cup \{0\}$, and the VC-Sauer Lemma (Vapnik and Chervonenkis, 1971; Sauer, 1972) implies $|\mathbb{C}[\mathcal{U}]| \leq \left(\frac{em}{d}\right)^d$, we have that for any $n \geq 2 \frac{\mathfrak{s} d}{\mathrm{Log}(\mathfrak{s})} \log_2\left(\frac{em}{d}\right)$, if $\forall j, f^\star_{\mathcal{P}_{XY}}(X_j) = Y_j$, then $\hat{h}_n(x) = f^\star_{\mathcal{P}_{XY}}(x)$ for every $x \in \mathcal{U}$. Thus, for $n \geq 2 \frac{\mathfrak{s} d}{\mathrm{Log}(\mathfrak{s})} \log_2\left(\frac{em}{d}\right)$, with probability one the classifier $\hat{h}_n$ returned by MEMB-HALVING-2 has $\hat{h}_n(x) = f^\star_{\mathcal{P}_{XY}}(x)$ for every $x \in \mathcal{U}$. Furthermore, as proven above, with probability at least $1 - \delta$, $\mathcal{U}$ is an ε-net of $\mathcal{P}$ for $\{\mathrm{DIS}(\{h, g\}) : h, g \in \mathbb{C}\}$. Thus, since $f^\star_{\mathcal{P}_{XY}}, \hat{h}_n \in \mathbb{C}$, by a union bound we have that for any $n \geq 2 \frac{\mathfrak{s} d}{\mathrm{Log}(\mathfrak{s})} \log_2\left(\frac{em}{d}\right)$, with probability at least $1 - \delta$, $\mathcal{P}(\mathrm{DIS}(\{f^\star_{\mathcal{P}_{XY}}, \hat{h}_n\})) \leq \varepsilon$. Since $\mathcal{P}_{XY} \in \mathrm{RE}$, this implies $\mathrm{er}_{\mathcal{P}_{XY}}(\hat{h}_n) = \mathcal{P}(\mathrm{DIS}(\{f^\star_{\mathcal{P}_{XY}}, \hat{h}_n\})) \leq \varepsilon$ as well. Thus, since this reasoning holds for any $\mathcal{P}_{XY} \in \mathrm{RE}$, we have established that

$$\Lambda_{\mathrm{RE}}(\varepsilon, \delta) \leq 2 \frac{\mathfrak{s} d}{\mathrm{Log}(\mathfrak{s})} \log_2\left(\frac{em}{d}\right) \leq 16 \mathrm{Log}\left(2ec'\right) \frac{\mathfrak{s} d}{\mathrm{Log}(\mathfrak{s})} \mathrm{Log}\left(\frac{1}{\varepsilon}\right).$$

14. The two cases not covered by the theorem of Hegedüs (1995) are the case $|\mathbb{C}[\mathcal{U}]| = 1$, for which the algorithm returns the sole element of $\mathbb{C}[\mathcal{U}]$ (which must agree with $f^\star_{\mathcal{P}_{XY}}$ on $\mathcal{U}$) without requesting any labels, and the case $|\mathbb{C}[\mathcal{U}]| = 2$, for which one can easily verify that $\mathrm{XTD}(\mathbb{C}[\mathcal{U}], \mathcal{U}) = 1$ and that the algorithm returns a classifier with the claimed property after requesting exactly one label.

Finally, we establish the $\mathfrak{s}\mathrm{Log}\left(\frac{1}{\varepsilon}\right)$ upper bound, as follows. Note that, since $|\mathbb{C}| \geq 2$, we must have $\mathfrak{s} \geq 1$. Fix any $\mathcal{P}_{XY} \in \mathrm{RE}$. Let $\mathcal{T} = \{\mathrm{DIS}(V_{S,h}) : S \in \bigcup_{m\in\mathbb{N}} \mathcal{X}^m, h \text{ a classifier}\}$, and for each $x_1, \ldots, x_{\mathfrak{s}} \in \mathcal{X}$ and $y_1, \ldots, y_{\mathfrak{s}} \in \mathcal{Y}$, define

$$\phi_{\mathfrak{s}}(x_1, \ldots, x_{\mathfrak{s}}, y_1, \ldots, y_{\mathfrak{s}}) = \mathrm{DIS}(\{g \in \mathbb{C} : \forall i \leq \mathfrak{s}, g(x_i) = y_i\}) \in \mathcal{T}.$$

Let $\tilde{c}'$ be as in Lemma 24, and define $\delta' = \delta/\left(2\lceil\log_2(1/\varepsilon)\rceil\right)$, $\ell = \lceil 2\tilde{c}'(\mathfrak{s}\mathrm{Log}(3\tilde{c}')+\mathrm{Log}(1/\delta'))\rceil$, $m = \lceil 2\tilde{c}'\mathfrak{s}\rceil$, and $\tilde{j} = \lceil(2m\lceil\log_2(2/\delta')\rceil + 2\ell)/\varepsilon\rceil$. Consider the following algorithm.

Algorithm 0
Input: label budget n
Output: classifier $\hat{h}_n$

---

0. $V_0 \leftarrow \mathbb{C}$, $\bar{j}_0 = 0$
1. For $k = 1, 2, \ldots, \lfloor n/m \rfloor$
2. If $|\{j \in \{\bar{j}_{k-1}+1, \ldots, \bar{j}_{k-1}+\tilde{j}\} : X_j \in \mathrm{DIS}(V_{k-1})\}| < m\lceil\log_2(2/\delta')\rceil + \ell$
3. Return any $\hat{h}_n \in V_{k-1}$ (or an arbitrary classifier $\hat{h}_n$ if $V_{k-1} = \emptyset$)
4. Let $j_{k,1}, \ldots, j_{k,m\lceil\log_2(2/\delta')\rceil+\ell}$ denote the $m\lceil\log_2(2/\delta')\rceil + \ell$ smallest indices in the set $\{j \in \{\bar{j}_{k-1}+1, \ldots, \bar{j}_{k-1}+\tilde{j}\} : X_j \in \mathrm{DIS}(V_{k-1})\}$ (in increasing order)
5. Let $\bar{j}_k = j_{k,m\lceil\log_2(2/\delta')\rceil+\ell}$
6. For each $i \in \mathbb{N}$, let

$$I_i = \left\{(i_1, \ldots, i_{\mathfrak{s}}, y_1, \ldots, y_{\mathfrak{s}}) \in \mathbb{N}^{\mathfrak{s}} \times \mathcal{Y}^{\mathfrak{s}} : m(i-1) < i_1 \leq \cdots \leq i_{\mathfrak{s}} \leq mi, \right.$$
$$\left. \sum_{t=m(i-1)+1}^{mi} \mathbb{1}_{\phi_{\mathfrak{s}}(X_{j_{k,i_1}}, \ldots, X_{j_{k,i_{\mathfrak{s}}}}, y_1, \ldots, y_{\mathfrak{s}})}(X_{j_{k,t}}) = 0 \right\}$$

7. Let

$$\hat{i}_k = \operatorname*{argmin}_{i \in \{1,\ldots,\lceil\log_2(2/\delta')\rceil\}} \max_{(i_1,\ldots,i_{\mathfrak{s}},y_1,\ldots,y_{\mathfrak{s}})\in I_i} \sum_{t=m\lceil\log_2(2/\delta')\rceil+1}^{m\lceil\log_2(2/\delta')\rceil+\ell} \mathbb{1}_{\phi_{\mathfrak{s}}(X_{j_{k,i_1}}, \ldots, X_{j_{k,i_{\mathfrak{s}}}}, y_1, \ldots, y_{\mathfrak{s}})}(X_{j_{k,t}})$$

8. Request the label $Y_{j_{k,t}}$ for each $t \in \left\{m\left(\hat{i}_k - 1\right) + 1, \ldots, m\hat{i}_k\right\}$
9. Let $V_k \leftarrow \left\{g \in V_{k-1} : \forall t \in \left\{m\left(\hat{i}_k - 1\right) + 1, \ldots, m\hat{i}_k\right\}, g(X_{j_{k,t}}) = Y_{j_{k,t}}\right\}$
10. Return any $\hat{h}_n \in V_{\lfloor n/m \rfloor}$

Fix any $k \in \{1, \ldots, \lfloor n/m \rfloor\}$. In the event that V_{k-1} is defined, let

$$M_k = \left|\left\{j \in \left\{\bar{j}_{k-1}+1, \ldots, \bar{j}_{k-1}+\tilde{j}\right\} : X_j \in \mathrm{DIS}(V_{k-1})\right\}\right|.$$

By a Chernoff bound (applied under the conditional distribution given V_{k-1} and $\bar{j}_{k-1}$) and the law of total probability (integrating out V_{k-1} and $\bar{j}_{k-1}$), there is an event E'_k of probability at least $1 - \delta'$, on which, if V_{k-1} is defined and satisfies

$$\mathcal{P}(\mathrm{DIS}(V_{k-1})) \geq 2\tilde{j}^{-1}\left(m\lceil\log_2(2/\delta')\rceil + \ell\right), \tag{10}$$

then $M_k \geq (1/2)\tilde{j}\mathcal{P}(\mathrm{DIS}(V_{k-1})) \geq m\lceil\log_2(2/\delta')\rceil + \ell$, in which case the algorithm will execute Steps 4-9 for this particular value of k, and in particular, the set V_k is defined. In this case, denote $\mathcal{U}_k = \left\{X_{j_{k,t}} : t \in \left\{m\left(\hat{i}_k - 1\right) + 1, \ldots, m\hat{i}_k\right\}\right\}$, which is well-defined in this case.

Next note that, on the event that V_{k-1} is defined, the M_k samples

$$\left\{X_j : j \in \left\{\bar{j}_{k-1} + 1, \ldots, \bar{j}_{k-1} + \tilde{j}\right\}, X_j \in \mathrm{DIS}(V_{k-1})\right\}$$

are conditionally independent given V_{k-1}, $\bar{j}_{k-1}$, and M_k, each having conditional distribution $\mathcal{P}(\cdot|\mathrm{DIS}(V_{k-1}))$. Thus, applying Lemma 24 under the conditional distribution given V_{k-1}, $\bar{j}_{k-1}$, and M_k, combined with the law of total probability (integrating out V_{k-1}, $\bar{j}_{k-1}$, and M_k), we have that there exists an event E_k of probability at least $1 - \delta'$, on which, if V_{k-1} is defined, and $M_k \geq m\lceil\log_2(2/\delta')\rceil + \ell$, then $\mathcal{U}_k$ is a $\frac{1}{2}$-net of $\mathcal{P}(\cdot|\mathrm{DIS}(V_{k-1}))$ for

$$\left\{\phi_{\mathfrak{s}}(X_{j_{k,i_1}}, \ldots, X_{j_{k,i_{\mathfrak{s}}}}, y_1, \ldots, y_{\mathfrak{s}}) : m\left(\hat{i}_k - 1\right) + 1 < i_1 \leq \cdots \leq i_{\mathfrak{s}} \leq m\hat{i}_k, y_1, \ldots, y_{\mathfrak{s}} \in \mathcal{Y}\right\}. \tag{11}$$

Together, we have that on $E_k \cap E_k'$, if V_{k-1} is defined and satisfies (10), then $\mathcal{U}_k$ is a $\frac{1}{2}$-net of $\mathcal{P}(\cdot|\mathrm{DIS}(V_{k-1}))$ for the collection (11).

In particular, Theorem 13 implies that, for any $x_1, \ldots, x_m \in \mathcal{X}^m$ and classifier $f \in \mathbb{C}$, $\exists i_1, \ldots, i_{\mathfrak{s}} \in \{1, \ldots, m\}$ such that $\{g \in \mathbb{C} : \forall j \leq \mathfrak{s}, g(x_{i_j}) = f(x_{i_j})\} = \{g \in \mathbb{C} : \forall i \leq m, g(x_i) = f(x_i)\}$ (see the discussion in Section 7.3.1), and since the left hand side is invariant to permutations of the i_j values, without loss of generality we may take $i_1 \leq \cdots \leq i_{\mathfrak{s}}$. This implies that on $E_k \cap E_k'$, if V_{k-1} is defined and satisfies (10), then $\exists i_1', \ldots, i_{\mathfrak{s}}' \in \left\{m\left(\hat{i}_k - 1\right) + 1, \ldots, m\hat{i}_k\right\}$ with $i_1' \leq \cdots \leq i_{\mathfrak{s}}'$ such that

$$\begin{aligned}
&\phi_{\mathfrak{s}}(X_{j_{k,i_1'}}, \ldots, X_{j_{k,i_{\mathfrak{s}}'}}, f(X_{j_{k,i_1'}}), \ldots, f(X_{j_{k,i_{\mathfrak{s}}'}})) \\
&= \mathrm{DIS}\left(\left\{g \in \mathbb{C} : \forall t \in \left\{m\left(\hat{i}_k - 1\right) + 1, \ldots, m\hat{i}_k\right\}, g(X_{j_{k,t}}) = f(X_{j_{k,t}})\right\}\right) = \mathrm{DIS}(V_{\mathcal{U}_k,f}),
\end{aligned}$$

so that

$$\begin{aligned}
&\mathrm{DIS}(V_{\mathcal{U}_k,f}) \in \\
&\quad\left\{\phi_{\mathfrak{s}}(X_{j_{k,i_1}}, \ldots, X_{j_{k,i_{\mathfrak{s}}}}, y_1, \ldots, y_{\mathfrak{s}}) : m\left(\hat{i}_k - 1\right) < i_1 \leq \cdots \leq i_{\mathfrak{s}} \leq m\hat{i}_k, y_1, \ldots, y_{\mathfrak{s}} \in \mathcal{Y}\right\}.
\end{aligned}$$

But we certainly have $\mathrm{DIS}(V_{\mathcal{U}_k,f}) \cap \mathcal{U}_k = \emptyset$. Thus, by the $\frac{1}{2}$-net property, on the event $E_k \cap E_k'$, if V_{k-1} is defined and satisfies (10), then every $f \in \mathbb{C}$ has

$$\mathcal{P}\left(\mathrm{DIS}(V_{\mathcal{U}_k,f})\middle|\mathrm{DIS}(V_{k-1})\right) \leq \frac{1}{2}. \tag{12}$$

Also note that, since $\mathcal{P}_{XY} \in \mathrm{RE}$, we have $f^\star_{\mathcal{P}_{XY}} \in \mathbb{C}$, and furthermore that there is an event E of probability one, on which $\forall j, Y_j = f^\star_{\mathcal{P}_{XY}}(X_j)$. In particular, on E, if V_{k-1} and V_k are defined, then $V_k = V_{\mathcal{U}_k, f^\star_{\mathcal{P}_{XY}}} \cap V_{k-1}$, which implies $\mathrm{DIS}(V_k) = \mathrm{DIS}\left(V_{\mathcal{U}_k, f^\star_{\mathcal{P}_{XY}}} \cap V_{k-1}\right) \subseteq$

$\mathrm{DIS}(V_{k-1})$. Thus, applying (12) with $f = f^\star_{\mathcal{P}_{XY}}$, we have that on the event $E \cap E_k \cap E'_k$, if V_{k-1} is defined and satisfies (10), then V_k is defined and satisfies

$$\begin{aligned}\mathcal{P}(\mathrm{DIS}(V_k)) &= \mathcal{P}(\mathrm{DIS}(V_k)|\mathrm{DIS}(V_{k-1}))\mathcal{P}(\mathrm{DIS}(V_{k-1})) \\ &\le \mathcal{P}\left(\mathrm{DIS}\left(V_{\mathcal{U}_k, f^\star_{\mathcal{P}_{XY}}}\right)\middle|\mathrm{DIS}(V_{k-1})\right)\mathcal{P}(\mathrm{DIS}(V_{k-1})) \le \frac{1}{2}\mathcal{P}(\mathrm{DIS}(V_{k-1})).\end{aligned}$$

Now suppose $\lfloor n/m \rfloor \ge \lceil \log_2(1/\varepsilon) \rceil$. Applying the above to every $k \le \lceil \log_2(1/\varepsilon) \rceil$, we have that there exist events E'_k and E_k for each $k \in \{1, \ldots, \lceil \log_2(1/\varepsilon) \rceil\}$, each of probability at least $1 - \delta'$, such that on the event $E \cap \bigcap_{k=1}^{\lceil \log_2(1/\varepsilon) \rceil} E'_k \cap E_k$, every $k \in \{1, \ldots, \lceil \log_2(1/\varepsilon) \rceil\}$ with V_{k-1} defined either has $\mathcal{P}(\mathrm{DIS}(V_{k-1})) < 2\tilde{j}^{-1}\left(m\lceil \log_2(2/\delta') \rceil + \ell\right)$ or else V_k is defined and satisfies $\mathcal{P}(\mathrm{DIS}(V_k)) \le \frac{1}{2}\mathcal{P}(\mathrm{DIS}(V_{k-1}))$. Since $V_0 = \mathbb{C}$ is defined, by induction we have that on the event $E \cap \bigcap_{k=1}^{\lceil \log_2(1/\varepsilon) \rceil} E'_k \cap E_k$, either some $k \in \{1, \ldots, \lceil \log_2(1/\varepsilon) \rceil\}$ has V_{k-1} defined and satisfies $\mathcal{P}(\mathrm{DIS}(V_{k-1})) < 2\tilde{j}^{-1}\left(m\lceil \log_2(2/\delta') \rceil + \ell\right)$, or else every $k \in \{1, \ldots, \lceil \log_2(1/\varepsilon) \rceil\}$ has V_k defined and satisfying $\mathcal{P}(\mathrm{DIS}(V_k)) \le \frac{1}{2}\mathcal{P}(\mathrm{DIS}(V_{k-1}))$. In particular, in this latter case, since $\mathcal{P}(\mathrm{DIS}(V_0)) \le 1$, by induction we have $\mathcal{P}(\mathrm{DIS}(V_{\lceil \log_2(1/\varepsilon) \rceil})) \le 2^{-\lceil \log_2(1/\varepsilon) \rceil} \le \varepsilon$.

Also note that $2\tilde{j}^{-1}\left(m\lceil \log_2(2/\delta') \rceil + \ell\right) \le \varepsilon$. Thus, denoting by $\hat{k}$ the largest $k \le \lfloor n/m \rfloor$ for which V_k is defined (which also implies V_k is defined for every $k \in \{0, \ldots, \hat{k}\}$), on the event $E \cap \bigcap_{k=1}^{\lceil \log_2(1/\varepsilon) \rceil} E'_k \cap E_k$, either some $k \le (\hat{k}+1) \wedge \lceil \log_2(1/\varepsilon) \rceil$ has $\mathcal{P}(\mathrm{DIS}(V_{k-1})) < \varepsilon$, so that (since $k \mapsto V_k$ is nonincreasing for $k \le \hat{k}$) $\mathcal{P}(\mathrm{DIS}(V_{\hat{k}})) \le \mathcal{P}(\mathrm{DIS}(V_{k-1})) < \varepsilon$, or else $\hat{k} \ge \lceil \log_2(1/\varepsilon) \rceil$, so that $\mathcal{P}(\mathrm{DIS}(V_{\hat{k}})) \le \mathcal{P}(\mathrm{DIS}(V_{\lceil \log_2(1/\varepsilon) \rceil})) \le \varepsilon$. Thus, on the event $E \cap \bigcap_{k=1}^{\lceil \log_2(1/\varepsilon) \rceil} E'_k \cap E_k$, in any case we have $\mathcal{P}(\mathrm{DIS}(V_{\hat{k}})) \le \varepsilon$. Furthermore, by the realizable case assumption, we have $f^\star_{\mathcal{P}_{XY}} \in V_0$, and if $f^\star_{\mathcal{P}_{XY}} \in V_{k-1}$ in Step 9, then (on the event E) $f^\star_{\mathcal{P}_{XY}} \in V_k$ as well. Thus, by induction, on the event E, $f^\star_{\mathcal{P}_{XY}} \in V_{\hat{k}}$. In particular, this also implies $V_{\hat{k}} \ne \emptyset$ on E, so that there exist valid choices of $\hat{h}_n$ in $V_{\hat{k}}$ upon reaching the "Return" step (Step 3, if $\hat{k} < \lfloor n/m \rfloor$, or Step 10, if $\hat{k} = \lfloor n/m \rfloor$). Thus, $\hat{h}_n \in V_{\hat{k}}$ as well on E, so that on the event E we have $\left\{x : \hat{h}_n(x) \ne f^\star_{\mathcal{P}_{XY}}(x)\right\} \subseteq \mathrm{DIS}(V_{\hat{k}})$. Therefore, on the event $E \cap \bigcap_{k=1}^{\lceil \log_2(1/\varepsilon) \rceil} E'_k \cap E_k$, we have

$$\mathrm{er}_{\mathcal{P}_{XY}}(\hat{h}_n) = \mathcal{P}\left(x : \hat{h}_n(x) \ne f^\star_{\mathcal{P}_{XY}}(x)\right) \le \mathcal{P}\left(\mathrm{DIS}\left(V_{\hat{k}}\right)\right) \le \varepsilon.$$

Finally, by a union bound, the event $E \cap \bigcap_{k=1}^{\lceil \log_2(1/\varepsilon) \rceil} E'_k \cap E_k$ has probability at least $1 - \lceil \log_2(1/\varepsilon) \rceil 2\delta' = 1 - \delta$. Noting that the above argument holds for any $\mathcal{P}_{XY} \in \mathrm{RE}$, and that the condition $\lfloor n/m \rfloor \ge \lceil \log_2(1/\varepsilon) \rceil$ is satisfied for any $n \ge 9\tilde{c}'\mathfrak{s}\mathrm{Log}(1/\varepsilon)$, this completes the proof that $\Lambda_{\mathrm{RE}}(\varepsilon, \delta) \le 9\tilde{c}'\mathfrak{s}\mathrm{Log}(1/\varepsilon) \lesssim \mathfrak{s}\mathrm{Log}(1/\varepsilon)$. ∎

B.2 The Noisy Cases

To extend the above ideas to noisy settings, we make use of a novel modification of a technique of Kääriäinen (2006). We first partition the data sequence into three parts. For $m \in \mathbb{N}$, let $X^1_m = X_{3(m-1)+1}$, $X^2_m = X_{3(m-1)+2}$, and let $X^3_m = X_{3m}$ and $Y^3_m = Y_{3m}$;

also denote $\mathbb{X}_1 = \{X_m^1\}_{m=1}^\infty$, $\mathbb{X}_2 = \{X_m^2\}_{m=1}^\infty$, $\mathbb{X}_3 = \{X_m^3\}_{m=1}^\infty$, $\mathbb{Y}_3 = \{Y_m^3\}_{m=1}^\infty$, and $\mathcal{Z} = \{(X_m, Y_m)\}_{m=1}^\infty$. Additionally, it will simplify some of the proofs to further partition $\mathbb{X}_3$ and $\mathbb{Y}_3$, as follows. Fix any bijection $\phi : \mathbb{N}^2 \to \mathbb{N}$, and for each $m, \ell \in \mathbb{N}$, let $X_{m,\ell}^3 = X_{\phi(m,\ell)}^3$ and $Y_{m,\ell}^3 = Y_{\phi(m,\ell)}^3$.

Fix values $\varepsilon, \delta \in (0,1)$, and let $\hat{\gamma}_\varepsilon$ be a value in $[\varepsilon/2, 1]$. Let $k_\varepsilon = \lceil \log_2(8/\hat{\gamma}_\varepsilon) \rceil$, and for each $k \in \{2, \ldots, k_\varepsilon\}$, define

$$\tilde{m}_k = \left\lceil \frac{16 \max\{c, 8\} k_\varepsilon}{2^k \varepsilon} \left(d\mathrm{Log}\left(\frac{2k_\varepsilon}{\varepsilon}\right) + \mathrm{Log}\left(\frac{64 k_\varepsilon}{\delta}\right)\right)\right\rceil,$$

for c as in Lemma 21. Also define $\tilde{m}_{k_\varepsilon+1} = 0$, $\tilde{m} = \tilde{m}_2$. and $q_{\varepsilon,\delta} = 2 + \left\lceil 2^{2k_\varepsilon+4} \ln\left(\frac{32\tilde{m}2^{2k_\varepsilon+3}}{\delta}\right)\right\rceil$. Also, for each $m \in \{1, \ldots, \tilde{m}\}$, define $\tilde{k}_m = \max\{k \in \{2, \ldots, k_\varepsilon\} : m \le \tilde{m}_k\}$ and let $\tilde{q}_m = 2^{3+2\tilde{k}_m} \ln(32\tilde{m} q_{\varepsilon,\delta}/\delta)$. Fix a value $\tau = \frac{\delta\varepsilon}{512\tilde{m}}$. Let $J_{\tau,\delta/2}$ be as in Lemma 27, applied to the sequence $X'_m = X_m^1$; to simplify notation, in this section we abbreviate $J = J_{\tau,\delta/2}$. Also, for each $x \in \mathcal{X}$, denote by $J(x)$ the (unique) set $A \in J$ with $x \in A$, and for each $m \in \{1, \ldots, \tilde{m}\}$, we abbreviate $J_m = J(X_m^2)$. Now consider the following algorithm.

Algorithm 1
Input: label budget n
Output: classifier $\hat{h}_n$

---

0. $V_0 \leftarrow \mathbb{C}$, $t \leftarrow 0$, $m \leftarrow 0$
1. While $t < n$ and $m < \tilde{m}$
2. $\quad m \leftarrow m + 1$
3. $\quad$ If $X_m^2 \in \mathrm{DIS}(V_{m-1})$
4. $\quad\quad$ Run Subroutine 1 with arguments $(n - t, m)$; let (q, y) be the returned values; let $t \leftarrow t + q$
5. $\quad\quad$ If $y \neq 0$ and $\exists h \in V_{m-1}$ with $h(X_m^2) = y$
6. $\quad\quad\quad$ Let $V_m \leftarrow \{h \in V_{m-1} : h(X_m^2) = y\}$
7. $\quad\quad$ Else let $V_m \leftarrow V_{m-1}$
8. $\quad$ Else let $V_m \leftarrow V_{m-1}$
9. Return any $\hat{h}_n \in V_m$

Subroutine 1
Input: label budget n, data point index m
Output: query counter q, value y

---

0. $\sigma_{m,0} \leftarrow 0$, $q \leftarrow 0$, $\ell_{m,0} \leftarrow 0$
1. Repeat
2. $\quad$ Let $\ell_{m,q+1} \leftarrow \min\{\ell > \ell_{m,q} : X_{m,\ell}^3 \in J_m\}$ (or $\ell_{m,q+1} \leftarrow 1$ if this set is empty)
3. $\quad$ Request the label $Y_{m,\ell_{m,q+1}}^3$; let $\sigma_{m,q+1} \leftarrow \sigma_{m,q} + Y_{m,\ell_{m,q+1}}^3$; let $q \leftarrow q + 1$
4. $\quad$ If $|\sigma_{m,q}| \ge 3\sqrt{2q \ln(32\tilde{m} q_{\varepsilon,\delta}/\delta)}$
5. $\quad\quad$ Return $(q, \mathrm{sign}(\sigma_{m,q}))$
6. $\quad$ Else if $q \ge \min\{n, \tilde{q}_m\}$
7. $\quad\quad$ Return $(q, 0)$

In this algorithm, the first part of the data (namely, $\mathbb{X}_1$) is used to partition the space via Lemma 27, so that each cell of the partition has $f^\star_{\mathcal{P}_{XY}}$ nearly-constant within it (assuming $f^\star_{\mathcal{P}_{XY}} \in \mathbb{C}$). The second part, $\mathbb{X}_2$, is used to simulate a commonly-studied active learning algorithm for the realizable case (namely, the algorithm of Cohn, Atlas, and Ladner, 1994), with two significant modifications. First, instead of directly requesting the label of a point, we use samples from the third part of the data (i.e., $\mathbb{X}_3$) that co-occur in the same cell of the partition as the would-be query point, repeatedly requesting for labels from that cell and using the majority vote of these returned labels in place of the label of the original point. Second, we discard a point X^2_m if we cannot identify a clear majority label within a certain number of queries, which decreases as the algorithm runs. Since this second modification often ends up rejecting more samples in cells with higher noise rates than those with lower noise rates, this effectively alters the marginal distribution over $\mathcal{X}$, shifting the distribution to favor less-noisy regions.

For the remainder of Appendix B.2, we fix an arbitrary probability measure $\mathcal{P}_{XY}$ over $\mathcal{X} \times \mathcal{Y}$ with $f^\star_{\mathcal{P}_{XY}} \in \mathbb{C}$, and as usual, we denote by $\mathcal{P}(\cdot) = \mathcal{P}_{XY}(\cdot \times \mathcal{Y})$ the marginal of $\mathcal{P}_{XY}$ over $\mathcal{X}$. For any $x \in \mathcal{X}$, define $\gamma_x = \left|\eta(x; \mathcal{P}_{XY}) - \frac{1}{2}\right|$, and define

$$\gamma_\varepsilon = \sup\left\{\gamma \in (0, 1/2] : \gamma \mathcal{P}(x : \gamma_x \leq \gamma) \leq \varepsilon/2\right\}.$$

Also, for the remainder of Appendix B.2, we suppose $\hat{\gamma}_\varepsilon$ is chosen to be in the range $[\varepsilon/2, \gamma_\varepsilon]$. For each $A \in J$, define

$$y_A = \operatorname*{argmax}_{y \in \mathcal{Y}} \mathcal{P}\left(x \in A : f^\star_{\mathcal{P}_{XY}}(x) = y\right) = \operatorname{sign}\left(\int_A f^\star_{\mathcal{P}_{XY}} \mathrm{d}\mathcal{P}\right),$$

and if $\mathcal{P}(A) > 0$, define $\eta(A; \mathcal{P}_{XY}) = \mathcal{P}_{XY}(A \times \{1\} | A \times \mathcal{Y})$ (i.e., the average value of $\eta(x; \mathcal{P}_{XY})$ over $x \in A$), and let $\gamma_A = \left|\eta(A; \mathcal{P}_{XY}) - \frac{1}{2}\right|$. For completeness, for any $A \in J$ with $\mathcal{P}(A) = 0$, define $\eta(A; \mathcal{P}_{XY}) = 1/2$ and $\gamma_A = 0$. Additionally, for each $n \in \mathbb{N} \cup \{\infty\}$ and $m \in \mathbb{N}$, let $(\hat{q}_{n,m}, \hat{y}_{n,m})$ denote the return values of Subroutine 1 when run with arguments (n, m).

Denote by E_1 the $\mathbb{X}_1$-measurable event of probability at least $1 - \delta/2$ implied by Lemma 27, on which every $h \in \mathbb{C}$ has

$$\sum_{A \in J} \min_{y \in \mathcal{Y}} \mathcal{P}\left(x \in A : h(x) = y\right) \leq \tau \tag{13}$$

and $\forall \gamma > 0$,

$$\mathcal{P}\left(\bigcup\left\{A \in J : \min_{y \in \mathcal{Y}} \mathcal{P}\left(x \in A : h(x) = y\right) > \gamma \mathcal{P}(A)\right\}\right) \leq \frac{\tau}{\gamma}. \tag{14}$$

We now proceed to characterize the behaviors of Subroutine 1 and Algorithm 1 via the following sequence of lemmas.

Lemma 28 *There exists a $(\mathbb{X}_1, \mathbb{X}_2, \mathbb{X}_3)$-measurable event E_0 of probability 1, on which $\forall m \in \{1, \ldots, \tilde{m}\}$, $\mathcal{P}(J_m) > 0$ and $|\{\ell \in \mathbb{N} : X^3_{m,\ell} \in J_m\}| = \infty$.*

Proof For each m, since each $A \in J$ with $\mathcal{P}(A) = 0$ has $\mathbb{P}(X_m^2 \in A) = 0$, and J has finite size, a union bound implies $\mathbb{P}(\mathcal{P}(J_m) = 0) = 0$. The strong law of large numbers (applied under the conditional distribution given J_m) and the law of total probability implies that $\frac{1}{\ell}\sum_{j=1}^{\ell} \mathbb{1}_{J_m}(X_{m,j}^3) \to \mathcal{P}(J_m)$ with probability 1, so that when $\mathcal{P}(J_m) > 0$, $\sum_{j=1}^{\ell} \mathbb{1}_{J_m}(X_{m,j}^3) \to \infty$. Finally, a union bound implies

$$\begin{aligned}
&\mathbb{P}\left(\exists m \le \tilde{m} : \mathcal{P}(J_m) = 0 \text{ or } |\{\ell \in \mathbb{N} : X_{m,\ell}^3 \in J_m\}| < \infty\right) \\
&\le \sum_{m=1}^{\tilde{m}} \mathbb{P}\left(\mathcal{P}(J_m) = 0\right) + \mathbb{P}\left(\mathcal{P}(J_m) > 0 \text{ and } |\{\ell \in \mathbb{N} : X_{m,\ell}^3 \in J_m\}| < \infty\right) = 0.
\end{aligned}$$

■

Lemma 29 *There exists a $(\mathbb{X}_1, \mathbb{X}_2)$-measurable event E_2 of probability at least $1 - \tau\tilde{m} \ge 1 - \delta/512$ such that, on $E_1 \cap E_2$, every $m \in \{1, \ldots, \tilde{m}\}$ has $f^\star_{\mathcal{P}_{XY}}(X_m^2) = y_{J_m}$.*

Proof Noting that, on E_1, (13) implies that

$$\begin{aligned}
\mathcal{P}\left(x : f^\star_{\mathcal{P}_{XY}}(x) \ne y_{J(x)}\right) &= \sum_{A \in J} \mathcal{P}\left(x \in A : f^\star_{\mathcal{P}_{XY}}(x) \ne y_A\right) \\
&= \sum_{A \in J} \min_{y \in \mathcal{Y}} \mathcal{P}\left(x \in A : f^\star_{\mathcal{P}_{XY}}(x) = y\right) \le \tau,
\end{aligned}$$

the result follows by a union bound. ■

Lemma 30 *There exists a $(\mathbb{X}_1, \mathbb{X}_2)$-measurable event E_3 of probability at least $1 - \frac{128\tau}{\varepsilon}\tilde{m} \ge 1 - \delta/4$ such that, on $E_1 \cap E_3$, every $m \in \{1, \ldots, \tilde{m}\}$ has $\mathcal{P}\left(x \in J_m : f^\star_{\mathcal{P}_{XY}}(x) \ne y_{J_m}\right) \le \frac{\varepsilon}{128}\mathcal{P}(J_m)$.*

Proof Noting that, on E_1, (14) implies that

$$\begin{aligned}
&\mathcal{P}\left(x : \mathcal{P}\left(x' \in J(x) : f^\star_{\mathcal{P}_{XY}}(x') \ne y_{J(x)}\right) > \frac{\varepsilon}{128}\mathcal{P}(J(x))\right) \\
&= \mathcal{P}\left(\bigcup\left\{A \in J : \mathcal{P}\left(x' \in A : f^\star_{\mathcal{P}_{XY}}(x') \ne y_A\right) > \frac{\varepsilon}{128}\mathcal{P}(A)\right\}\right) \\
&= \mathcal{P}\left(\bigcup\left\{A \in J : \min_{y \in \mathcal{Y}} \mathcal{P}\left(x' \in A : f^\star_{\mathcal{P}_{XY}}(x') = y\right) > \frac{\varepsilon}{128}\mathcal{P}(A)\right\}\right) \le \frac{128\tau}{\varepsilon},
\end{aligned}$$

the result follows by a union bound. ■

Lemma 31 *$\forall A \in J$,*

$$\mathcal{P}_{XY}\left(A \times \{y_A\}\right) \ge \frac{1}{2}\mathcal{P}(A) + \int_A \gamma_x \mathcal{P}(\mathrm{d}x) - \mathcal{P}\left(x \in A : f^\star_{\mathcal{P}_{XY}}(x) \ne y_A\right).$$

Proof Any $A \in J$ has

$$\begin{aligned}\mathcal{P}_{XY}\left(A \times \{y_A\}\right) &\geq \int_A \mathbb{1}[f^\star_{\mathcal{P}_{XY}}(x) = y_A]\left(\frac{1}{2} + \gamma_x\right)\mathcal{P}(\mathrm{d}x) \\ &\geq \int_A \left(\frac{1}{2} + \gamma_x\right)\mathcal{P}(\mathrm{d}x) - \mathcal{P}\left(x \in A : f^\star_{\mathcal{P}_{XY}}(x) \neq y_A\right) \\ &= \frac{1}{2}\mathcal{P}(A) + \int_A \gamma_x \mathcal{P}(\mathrm{d}x) - \mathcal{P}\left(x \in A : f^\star_{\mathcal{P}_{XY}}(x) \neq y_A\right).\end{aligned}$$

■

Lemma 32 *On the event $E_0 \cap E_1 \cap E_3$, every $m \in \{1, \ldots, \tilde{m}\}$ with $\gamma_{J_m} > \varepsilon/128$ has $\mathcal{P}_{XY}(J_m \times \{y_{J_m}\}) > \mathcal{P}_{XY}(J_m \times \{-y_{J_m}\})$, and every $m \in \{1, \ldots, \tilde{m}\}$ with $\int_{J_m} \gamma_x \mathcal{P}(\mathrm{d}x) > (\varepsilon/2)\mathcal{P}(J_m)$ has*

$$\int_{J_m} \gamma_x \mathcal{P}(\mathrm{d}x) \geq \gamma_{J_m}\mathcal{P}(J_m) \geq \frac{63}{64}\int_{J_m} \gamma_x \mathcal{P}(\mathrm{d}x) > \frac{63}{128}\varepsilon\mathcal{P}(J_m). \tag{15}$$

Proof Jensen's inequality implies we always have $\gamma_A \mathcal{P}(A) \leq \int_A \gamma_x \mathcal{P}(\mathrm{d}x)$. In particular, this implies that any $A \in J$ with $\mathcal{P}(A) > 0$ and $\mathcal{P}(x \in A : f^\star_{\mathcal{P}_{XY}}(x) \neq y_A) \leq \frac{\varepsilon}{128}\mathcal{P}(A)$ and $\gamma_A > \varepsilon/128$ has $\int_A \gamma_x \mathcal{P}(\mathrm{d}x) - \mathcal{P}(x \in A : f^\star_{\mathcal{P}_{XY}}(x) \neq y_A) \geq \gamma_A \mathcal{P}(A) - \mathcal{P}(x \in A : f^\star_{\mathcal{P}_{XY}}(x) \neq y_A) > (\varepsilon/128)\mathcal{P}(A) - (\varepsilon/128)\mathcal{P}(A) = 0$, so that Lemma 31 implies $\mathcal{P}_{XY}(A \times \{y_A\}) > \frac{1}{2}\mathcal{P}(A)$, and therefore $\mathcal{P}_{XY}(A \times \{y_A\}) > \mathcal{P}_{XY}(A \times \{-y_A\})$. Since Lemmas 28 and 30 imply that, on $E_0 \cap E_1 \cap E_3$, for every $m \in \{1, \ldots, \tilde{m}\}$, $\mathcal{P}(J_m) > 0$ and $\mathcal{P}(x \in J_m : f^\star_{\mathcal{P}_{XY}}(x) \neq y_{J_m}) \leq \frac{\varepsilon}{128}\mathcal{P}(J_m)$, we have established the first claim in the lemma statement.

For the second claim, the first inequality follows by Jensen's inequality. For the second inequality, note that any $A \in J$ has $\gamma_A \mathcal{P}(A) \geq \mathcal{P}_{XY}(A \times \{y_A\}) - \frac{1}{2}\mathcal{P}(A)$, so that Lemma 31 implies $\gamma_A \mathcal{P}(A) \geq \int_A \gamma_x \mathcal{P}(\mathrm{d}x) - \mathcal{P}(x \in A : f^\star_{\mathcal{P}_{XY}}(x) \neq y_A)$. Therefore, since Lemma 30 implies that, on $E_1 \cap E_3$, every $m \in \{1, \ldots, \tilde{m}\}$ has $\mathcal{P}(x \in J_m : f^\star_{\mathcal{P}_{XY}}(x) \neq y_{J_m}) \leq \frac{\varepsilon}{128}\mathcal{P}(J_m)$, we have that on $E_1 \cap E_3$, any $m \in \{1, \ldots, \tilde{m}\}$ with $\int_{J_m} \gamma_x \mathcal{P}(\mathrm{d}x) > (\varepsilon/2)\mathcal{P}(J_m)$ has $\mathcal{P}(x \in J_m : f^\star_{\mathcal{P}_{XY}}(x) \neq y_{J_m}) \leq \frac{1}{64}\int_{J_m} \gamma_x \mathcal{P}(\mathrm{d}x)$, so that $\gamma_{J_m}\mathcal{P}(J_m) \geq \int_{J_m} \gamma_x \mathcal{P}(\mathrm{d}x) - \mathcal{P}(x \in J_m : f^\star_{\mathcal{P}_{XY}}(x) \neq y_{J_m}) \geq \frac{63}{64}\int_{J_m} \gamma_x \mathcal{P}(\mathrm{d}x)$. The final inequality then follows by the assumption that $\int_{J_m} \gamma_x \mathcal{P}(\mathrm{d}x) > (\varepsilon/2)\mathcal{P}(J_m)$. ■

Lemma 33 *On E_1, $\forall \gamma > (1/4)\gamma_\varepsilon$,*

$$\mathcal{P}\left(\bigcup\{A \in J : \gamma_A \leq \gamma\}\right) \leq 3\mathcal{P}(x : \gamma_x < 4\gamma),$$

and $\forall \gamma \in (0, (1/4)\gamma_\varepsilon]$,

$$\mathcal{P}\left(\bigcup\{A \in J : \gamma_A \leq \gamma\}\right) \leq \frac{3\varepsilon}{2\gamma_\varepsilon}.$$

Proof By Markov's inequality, for any $\gamma > 0$, any $A \in J$ with $\int_A \gamma_x \mathcal{P}(\mathrm{d}x) \leq \gamma \mathcal{P}(A)$ must have $\mathcal{P}(x \in A : \gamma_x \geq 2\gamma) \leq \frac{1}{2}\mathcal{P}(A)$, which implies $\mathcal{P}(x \in A : \gamma_x < 2\gamma) \geq \frac{1}{2}\mathcal{P}(A)$. Therefore,

$$\mathcal{P}\left(\bigcup\left\{A \in J : \int_A \gamma_x \mathcal{P}(\mathrm{d}x) \leq \gamma \mathcal{P}(A)\right\}\right) \leq \mathcal{P}\left(\bigcup\left\{A \in J : \mathcal{P}(x \in A : \gamma_x < 2\gamma) \geq \frac{1}{2}\mathcal{P}(A)\right\}\right) \leq 2\mathcal{P}(x : \gamma_x < 2\gamma), \quad (16)$$

where the last inequality is due to Markov's inequality.

Also, for every $\gamma > 0$, since $\gamma_A \mathcal{P}(A) \geq \mathcal{P}_{XY}(A \times \{y_A\}) - \frac{1}{2}\mathcal{P}(A)$,

$$\mathcal{P}\left(\bigcup\{A \in J : \gamma_A \leq \gamma\}\right) = \mathcal{P}\left(\bigcup\{A \in J : \gamma_A \mathcal{P}(A) \leq \gamma \mathcal{P}(A)\}\right) \leq \mathcal{P}\left(\bigcup\left\{A \in J : \mathcal{P}_{XY}(A \times \{y_A\}) - \frac{1}{2}\mathcal{P}(A) \leq \gamma \mathcal{P}(A)\right\}\right).$$

Lemma 31 implies $\mathcal{P}_{XY}(A \times \{y_A\}) - \frac{1}{2}\mathcal{P}(A) \geq \int_A \gamma_x \mathcal{P}(\mathrm{d}x) - \mathcal{P}(x \in A : f^\star_{\mathcal{P}_{XY}}(x) \neq y_A)$, so that the above is at most

$$\mathcal{P}\left(\bigcup\left\{A \in J : \int_A \gamma_x \mathcal{P}(\mathrm{d}x) \leq \gamma \mathcal{P}(A) + \mathcal{P}(x \in A : f^\star_{\mathcal{P}_{XY}}(x) \neq y_A)\right\}\right).$$

By a union bound, this is at most

$$\mathcal{P}\left(\bigcup\left\{A \in J : \int_A \gamma_x \mathcal{P}(\mathrm{d}x) \leq 2\gamma \mathcal{P}(A)\right\}\right) + \mathcal{P}\left(\bigcup\left\{A \in J : \mathcal{P}(x \in A : f^\star_{\mathcal{P}_{XY}}(x) \neq y_A) > \gamma \mathcal{P}(A)\right\}\right). \quad (17)$$

On E_1, (14) implies that

$$\mathcal{P}\left(\bigcup\left\{A \in J : \mathcal{P}(x \in A : f^\star_{\mathcal{P}_{XY}}(x) \neq y_A) > \gamma \mathcal{P}(A)\right\}\right) \leq \frac{\tau}{\gamma} < \frac{\varepsilon}{8\gamma}.$$

Furthermore, by (16),

$$\mathcal{P}\left(\bigcup\left\{A \in J : \int_A \gamma_x \mathcal{P}(\mathrm{d}x) \leq 2\gamma \mathcal{P}(A)\right\}\right) \leq 2\mathcal{P}(x : \gamma_x < 4\gamma).$$

Using these two inequalities to bound the two terms in (17), we have that

$$\mathcal{P}\left(\bigcup\{A \in J : \gamma_A \leq \gamma\}\right) \leq 2\mathcal{P}(x : \gamma_x < 4\gamma) + \frac{\varepsilon}{8\gamma}.$$

By definition of γ_ε, if $\gamma > (1/4)\gamma_\varepsilon$, we must have $4\gamma\mathcal{P}(x : \gamma_x < 4\gamma) \geq \gamma_\varepsilon \mathcal{P}(x : \gamma_x \leq \gamma_\varepsilon) \geq \varepsilon/2$, so that $\frac{\varepsilon}{8\gamma} \leq \mathcal{P}(x : \gamma_x < 4\gamma)$, which implies

$$2\mathcal{P}(x : \gamma_x < 4\gamma) + \frac{\varepsilon}{8\gamma} \leq 3\mathcal{P}(x : \gamma_x < 4\gamma),$$

which establishes the first claim. On the other hand, if $0 < \gamma \leq (1/4)\gamma_\varepsilon$, we have $4\gamma\mathcal{P}(x : \gamma_x < 4\gamma) \leq \varepsilon/2$, so that $2\mathcal{P}(x : \gamma_x < 4\gamma) \leq \frac{\varepsilon}{4\gamma}$, which implies

$$2\mathcal{P}(x : \gamma_x < 4\gamma) + \frac{\varepsilon}{8\gamma} \leq \frac{3\varepsilon}{8\gamma}.$$

This establishes the second claim, since (combined with monotonicity of probabilities) it implies

$$\mathcal{P}\left(\bigcup\{A \in J : \gamma_A \leq \gamma\}\right) \leq \mathcal{P}\left(\bigcup\{A \in J : \gamma_A \leq (1/4)\gamma_\varepsilon\}\right) \leq \frac{3\varepsilon}{2\gamma_\varepsilon}.$$

■

Lemma 34 *On E_1, $\forall h \in \mathbb{C}$,*

$$\mathrm{er}_{\mathcal{P}_{XY}}(h) - \mathrm{er}_{\mathcal{P}_{XY}}(f^\star_{\mathcal{P}_{XY}}) \leq 5\tau + \int \mathbb{1}[h(x) \neq f^\star_{\mathcal{P}_{XY}}(x)]2\gamma_{J(x)}\mathcal{P}(\mathrm{d}x).$$

Proof For any $h \in \mathbb{C}$, we generally have

$$\mathrm{er}_{\mathcal{P}_{XY}}(h) - \mathrm{er}_{\mathcal{P}_{XY}}(f^\star_{\mathcal{P}_{XY}}) = \int \mathbb{1}[h(x) \neq f^\star_{\mathcal{P}_{XY}}(x)]2\gamma_x\mathcal{P}(\mathrm{d}x).$$

For each $A \in J$, let $y^h_A = \mathrm{argmax}_{y\in\mathcal{Y}}\,\mathcal{P}(x : h(x) = y)$. $\forall x \in \mathcal{X}$, $\mathbb{1}[h(x) \neq f^\star_{\mathcal{P}_{XY}}(x)]2\gamma_x \leq 1$. Therefore,

$$\int \mathbb{1}[h(x) \neq f^\star_{\mathcal{P}_{XY}}(x)]2\gamma_x\mathcal{P}(\mathrm{d}x) \leq \mathcal{P}\left(x : h(x) \neq y^h_{J(x)} \text{ or } f^\star_{\mathcal{P}_{XY}}(x) \neq y_{J(x)}\right) + \int_{\left\{x:h(x)=y^h_{J(x)}, f^\star_{\mathcal{P}_{XY}}(x)=y_{J(x)}\right\}} \mathbb{1}[y^h_{J(x)} \neq y_{J(x)}]2\gamma_x\mathcal{P}(\mathrm{d}x). \quad (18)$$

By a union bound,

$$\mathcal{P}\left(x : h(x) \neq y^h_{J(x)} \text{ or } f^\star_{\mathcal{P}_{XY}}(x) \neq y_{J(x)}\right) \leq \mathcal{P}\left(x : h(x) \neq y^h_{J(x)}\right) + \mathcal{P}\left(x : f^\star_{\mathcal{P}_{XY}}(x) \neq y_{J(x)}\right).$$

Furthermore, on E_1, (13) implies the right hand side is at most 2τ. Combining this with (18) implies

$$\mathrm{er}_{\mathcal{P}_{XY}}(h) - \mathrm{er}_{\mathcal{P}_{XY}}(f^\star_{\mathcal{P}_{XY}}) \leq 2\tau + \int_{\left\{x:h(x)=y^h_{J(x)}, f^\star_{\mathcal{P}_{XY}}(x)=y_{J(x)}\right\}} \mathbb{1}[y^h_{J(x)} \neq y_{J(x)}]2\gamma_x\mathcal{P}(\mathrm{d}x). \quad (19)$$

Also,

$$\int_{\left\{x:h(x)=y^h_{J(x)}, f^\star_{\mathcal{P}_{XY}}(x)=y_{J(x)}\right\}} \mathbb{1}[y^h_{J(x)} \neq y_{J(x)}]2\gamma_x\mathcal{P}(\mathrm{d}x)$$
$$= \sum_{A\in J: y^h_A \neq y_A} \int_{\left\{x\in A: h(x)=y^h_A, f^\star_{\mathcal{P}_{XY}}(x)=y_A\right\}} 2\gamma_x\mathcal{P}(\mathrm{d}x) \leq \sum_{A\in J: y^h_A \neq y_A} \int_{\left\{x\in A: f^\star_{\mathcal{P}_{XY}}(x)=y_A\right\}} 2\gamma_x\mathcal{P}(\mathrm{d}x).$$

Since $f^\star_{\mathcal{P}_{XY}}(x) = \text{sign}(2\eta(x;\mathcal{P}_{XY})-1)$ for every $x \in \mathcal{X}$, any measurable $C \subseteq \mathcal{X}$ has

$$\mathcal{P}_{XY}\left((x,y) : x \in C, y = f^\star_{\mathcal{P}_{XY}}(x)\right) = \int_C \left(\frac{1}{2} + \gamma_x\right) \mathcal{P}(\mathrm{d}x).$$

Therefore, for each $A \in J$,

$$\begin{aligned}
\gamma_A \mathcal{P}(A) &\geq \mathcal{P}_{XY}(A \times \{y_A\}) - \frac{1}{2}\mathcal{P}(A) \geq \mathcal{P}_{XY}\left(\left\{x \in A : f^\star_{\mathcal{P}_{XY}}(x) = y_A\right\} \times \{y_A\}\right) - \frac{1}{2}\mathcal{P}(A) \\
&= \int_{\left\{x \in A : f^\star_{\mathcal{P}_{XY}}(x) = y_A\right\}} \left(\frac{1}{2} + \gamma_x\right) \mathcal{P}(\mathrm{d}x) - \frac{1}{2}\mathcal{P}(A) \\
&= \int_{\left\{x \in A : f^\star_{\mathcal{P}_{XY}}(x) = y_A\right\}} \gamma_x \mathcal{P}(\mathrm{d}x) - \frac{1}{2}\mathcal{P}\left(x \in A : f^\star_{\mathcal{P}_{XY}}(x) \neq y_A\right).
\end{aligned}$$

Therefore,

$$\sum_{A \in J : y^h_A \neq y_A} \int_{\left\{x \in A : f^\star_{\mathcal{P}_{XY}}(x) = y_A\right\}} 2\gamma_x \mathcal{P}(\mathrm{d}x) \leq \sum_{A \in J : y^h_A \neq y_A} \mathcal{P}\left(x \in A : f^\star_{\mathcal{P}_{XY}}(x) \neq y_A\right) + 2\gamma_A \mathcal{P}(A).$$

On E_1, (13) implies that the right hand side is at most

$$\tau + \sum_{A \in J : y^h_A \neq y_A} 2\gamma_A \mathcal{P}(A).$$

Combining this with (19), we have that on E_1,

$$\mathrm{er}_{\mathcal{P}_{XY}}(h) - \mathrm{er}_{\mathcal{P}_{XY}}(f^\star_{\mathcal{P}_{XY}}) \leq 3\tau + \sum_{A \in J : y^h_A \neq y_A} 2\gamma_A \mathcal{P}(A). \tag{20}$$

For each $A \in J$ and $x \in A$, if $y^h_A \neq y_A$, then either $h(x) \neq f^\star_{\mathcal{P}_{XY}}(x)$ holds, or else one of $h(x) \neq y^h_A$ or $f^\star_{\mathcal{P}_{XY}}(x) \neq y_A$ holds. Thus, any $A \in J$ with $y^h_A \neq y_A$ has

$$\begin{aligned}
\mathcal{P}(A) &\leq \int_A \left(\mathbb{1}\left[h(x) \neq f^\star_{\mathcal{P}_{XY}}(x)\right] + \mathbb{1}\left[h(x) \neq y^h_A\right] + \mathbb{1}\left[f^\star_{\mathcal{P}_{XY}}(x) \neq y_A\right]\right) \mathcal{P}(\mathrm{d}x) \\
&= \mathcal{P}\left(x \in A : h(x) \neq y^h_A\right) + \mathcal{P}\left(x \in A : f^\star_{\mathcal{P}_{XY}}(x) \neq y_A\right) + \int_A \mathbb{1}\left[h(x) \neq f^\star_{\mathcal{P}_{XY}}(x)\right] \mathcal{P}(\mathrm{d}x).
\end{aligned}$$

Combined with (20), this implies that on E_1,

$$\begin{aligned}
&\mathrm{er}_{\mathcal{P}_{XY}}(h) - \mathrm{er}_{\mathcal{P}_{XY}}(f^\star_{\mathcal{P}_{XY}}) \\
&\quad \leq 3\tau + \sum_{A \in J : y^h_A \neq y_A} 2\gamma_A \Bigg(\mathcal{P}\left(x \in A : h(x) \neq y^h_A\right) + \mathcal{P}\left(x \in A : f^\star_{\mathcal{P}_{XY}}(x) \neq y_A\right) \\
&\qquad\qquad + \int_A \mathbb{1}\left[h(x) \neq f^\star_{\mathcal{P}_{XY}}(x)\right] \mathcal{P}(\mathrm{d}x)\Bigg).
\end{aligned}$$

Since $2\gamma_A \leq 1$, the right hand side is at most

$$3\tau + \sum_{A\in J} \mathcal{P}\left(x \in A : h(x) \neq y_A^h\right) + \sum_{A\in J} \mathcal{P}\left(x \in A : f^\star_{\mathcal{P}_{XY}}(x) \neq y_A\right) + \sum_{A\in J: y_A^h \neq y_A} 2\gamma_A \int_A \mathbb{1}\left[h(x) \neq f^\star_{\mathcal{P}_{XY}}(x)\right] \mathcal{P}(\mathrm{d}x),$$

and on E_1, (13) implies this is at most

$$5\tau + \sum_{A\in J: y_A^h \neq y_A} 2\gamma_A \int_A \mathbb{1}\left[h(x) \neq f^\star_{\mathcal{P}_{XY}}(x)\right] \mathcal{P}(\mathrm{d}x)$$
$$\leq 5\tau + \sum_{A\in J} \int_A \mathbb{1}\left[h(x) \neq f^\star_{\mathcal{P}_{XY}}(x)\right] 2\gamma_A \mathcal{P}(\mathrm{d}x) = 5\tau + \int \mathbb{1}\left[h(x) \neq f^\star_{\mathcal{P}_{XY}}(x)\right] 2\gamma_{J(x)} \mathcal{P}(\mathrm{d}x).$$

■

Lemma 35 *There is a $\mathcal{Z}$-measurable event E_4 of probability at least $1-\delta/32$ such that, on $\bigcap_{j=0}^4 E_j$, $\forall k \in \{2,\ldots,k_\varepsilon\}$, $\forall m \in \{\tilde{m}_{k+1}+1,\ldots,\tilde{m}_k\}$, $\forall n \in \mathbb{N}\cup\{\infty\}$, $\hat{y}_{n,m} \in \{0, f^\star_{\mathcal{P}_{XY}}(X^2_m)\}$, $\hat{q}_{n,m} \leq \left\lceil \frac{8}{\max\{\gamma^2_{J_m}, 2^{-2k}\}} \ln\left(\frac{32\tilde{m}q_{\varepsilon,\delta}}{\delta}\right)\right\rceil$, and if $\gamma_{J_m} \geq 2^{-k}$ then $\hat{y}_{\infty,m} = f^\star_{\mathcal{P}_{XY}}(X^2_m)$.*

Proof Since $\hat{q}_{n,m} \leq \hat{q}_{\infty,m}$, and $\hat{y}_{n,m} = 0$ whenever $\hat{q}_{n,m} < \hat{q}_{\infty,m}$, it suffices to show the claims hold for $\hat{q}_{\infty,m}$ and $\hat{y}_{\infty,m}$ for each $m \in \{1,\ldots,\tilde{m}\}$.

For each $m \in \{1,\ldots,\tilde{m}\}$, let $\ell_{m,1}, \ell_{m,2}, \ldots$ denote the increasing infinite subsequence of values $\ell \in \mathbb{N}$ with $X^3_{m,\ell} \in J_m$, guaranteed to exist by Lemma 28 on E_0; also, for each $q \in \mathbb{N}$, define $\sigma_{m,q} = \sum_{j=1}^q Y^3_{m,\ell_{m,j}}$. Note that these definitions of $\ell_{m,q}$ and $\sigma_{m,q}$ agree with those defined in Subroutine 1 for each $q \leq \hat{q}_{\infty,m}$. Let E_4 denote the event that E_0 occurs and that $\forall m \in \{1,\ldots,\tilde{m}\}$, $\forall q \in \{1,\ldots,q_{\varepsilon,\delta}\}$,

$$|\sigma_{m,q} - q(2\eta(J_m;\mathcal{P}_{XY}) - 1)| \leq \sqrt{2q \ln\left(\frac{32\tilde{m}q_{\varepsilon,\delta}}{\delta}\right)}. \tag{21}$$

For each $m \in \{1,\ldots,\tilde{m}\}$ and $q \in \{1,\ldots,q_{\varepsilon,\delta}\}$, Lemma 28 and Hoeffding's inequality imply that (21) holds with conditional probability (given J_m) at least $1 - \delta/(32\tilde{m}q_{\varepsilon,\delta})$. The law of total probability and a union bound over values of m and q then imply that E_4 has probability at least $1-\delta/32$.

Now fix any $k \in \{2,\ldots,k_\varepsilon\}$ and $m \in \{\tilde{m}_{k+1}+1,\ldots,\tilde{m}_k\}$. Since $\tilde{k}_m = k$, the condition in Step 6 guarantees $\hat{q}_{\infty,m} \leq \left\lceil 2^{2k+3} \ln\left(\frac{32\tilde{m}q_{\varepsilon,\delta}}{\delta}\right)\right\rceil$. Furthermore, if $\gamma_{J_m} \geq 2^{-k}$, then for

$$q = \left\lceil \frac{8}{\gamma^2_{J_m}} \ln\left(\frac{32\tilde{m}q_{\varepsilon,\delta}}{\delta}\right)\right\rceil,$$

we have

$$2q\gamma_m \geq 4\sqrt{2q \ln\left(\frac{32\tilde{m}q_{\varepsilon,\delta}}{\delta}\right)}.$$

In particular, recalling that $2q\gamma_{J_m} = |q(2\eta(J_m;\mathcal{P}_{XY})-1)|$, we have

$$|q(2\eta(J_m;\mathcal{P}_{XY})-1)| \geq 4\sqrt{2q\ln\left(\frac{32\tilde{m}q_{\varepsilon,\delta}}{\delta}\right)}. \tag{22}$$

Since $q_{\varepsilon,\delta} \geq \left\lceil 2^{2k+3}\ln\left(\frac{32\tilde{m}q_{\varepsilon,\delta}}{\delta}\right)\right\rceil \geq q$, the event E_4 implies that (21) holds, so that

$$\sigma_{m,q} \geq q(2\eta(J_m;\mathcal{P}_{XY})-1) - \sqrt{2q\ln\left(\frac{32\tilde{m}q_{\varepsilon,\delta}}{\delta}\right)}.$$

Thus, if $q(2\eta(J_m;\mathcal{P}_{XY})-1) \geq 4\sqrt{2q\ln\left(\frac{32\tilde{m}q_{\varepsilon,\delta}}{\delta}\right)}$, the condition in Step 4 will imply $\hat{q}_{\infty,m} \leq q$, and since $q \leq \tilde{q}_m$, that $\hat{y}_{\infty,m} \in \mathcal{Y}$. Likewise, (21) implies

$$\sigma_{m,q} \leq q(2\eta(J_m;\mathcal{P}_{XY})-1) + \sqrt{2q\ln\left(\frac{32\tilde{m}q_{\varepsilon,\delta}}{\delta}\right)},$$

so that $q(2\eta(J_m;\mathcal{P}_{XY})-1) \leq -4\sqrt{2q\ln\left(\frac{32\tilde{m}q_{\varepsilon,\delta}}{\delta}\right)}$ would also suffice to imply $\hat{q}_{\infty,m} \leq q$ and $\hat{y}_{\infty,m} \in \mathcal{Y}$ via the condition in Step 4. Thus, since (22) implies one of these two conditions holds, we have that on E_4, if $\gamma_{J_m} \geq 2^{-k}$ then $\hat{q}_{\infty,m} \leq \left\lceil \frac{8}{\gamma_{J_m}^2}\ln\left(\frac{32\tilde{m}q_{\varepsilon,\delta}}{\delta}\right)\right\rceil$ and $\hat{y}_{\infty,m} \in \mathcal{Y}$.

It remains only to show that $\hat{y}_{\infty,m} \in \{0, f^\star_{\mathcal{P}_{XY}}(X_m^2)\}$. This clearly holds if the return value originates in Step 7, so we need only consider the case where Subroutine 1 reaches Step 5. Due to the condition in Step 6, this cannot occur for a value of $q > q_{\varepsilon,\delta}$ (since $\tilde{q}_m \leq \tilde{q}_1 \leq q_{\varepsilon,\delta}$), so let us consider any value of $q \in \{1,\ldots,q_{\varepsilon,\delta}\}$, and suppose $|\sigma_{m,q}| \geq 3\sqrt{2q\ln\left(\frac{32\tilde{m}q_{\varepsilon,\delta}}{\delta}\right)}$. On the event E_4, (21) implies that if $\sigma_{m,q} \geq 3\sqrt{2q\ln\left(\frac{32\tilde{m}q_{\varepsilon,\delta}}{\delta}\right)}$, then $q(2\eta(J_m;\mathcal{P}_{XY})-1) \geq \sigma_{m,q} - \sqrt{2q\ln\left(\frac{32\tilde{m}q_{\varepsilon,\delta}}{\delta}\right)} \geq 2\sqrt{2q\ln\left(\frac{32\tilde{m}q_{\varepsilon,\delta}}{\delta}\right)} > 0$, and likewise if $\sigma_{m,q} \leq -3\sqrt{2q\ln\left(\frac{32\tilde{m}q_{\varepsilon,\delta}}{\delta}\right)}$, then $q(2\eta(J_m;\mathcal{P}_{XY})-1) \leq \sigma_{m,q} + \sqrt{2q\ln\left(\frac{32\tilde{m}q_{\varepsilon,\delta}}{\delta}\right)} \leq -2\sqrt{2q\ln\left(\frac{32\tilde{m}q_{\varepsilon,\delta}}{\delta}\right)} < 0$; thus, since $|2\eta(J_m;\mathcal{P}_{XY})-1| = 2\gamma_{J_m}$, if $|\sigma_{m,q}| \geq 3\sqrt{2q\ln\left(\frac{32\tilde{m}q_{\varepsilon,\delta}}{\delta}\right)}$, then

$$\gamma_{J_m} \geq \sqrt{\frac{2}{q}\ln\left(\frac{32\tilde{m}q_{\varepsilon,\delta}}{\delta}\right)} \tag{23}$$

and $\mathrm{sign}(2\eta(J_m;\mathcal{P}_{XY})-1) = \mathrm{sign}(\sigma_{m,q})$. In particular, since $q \leq q_{\varepsilon,\delta} \leq 2^{2k_\varepsilon+5}\ln\left(\frac{32\tilde{m}q_{\varepsilon,\delta}}{\delta}\right)$, this implies

$$\gamma_{J_m} \geq \sqrt{\frac{2}{q_{\varepsilon,\delta}}\ln\left(\frac{32\tilde{m}q_{\varepsilon,\delta}}{\delta}\right)} \geq 2^{-k_\varepsilon-2} > \varepsilon/128.$$

Therefore, Lemma 32 implies that on $\bigcap_{j=0}^4 E_j$, $\mathrm{sign}(2\eta(J_m;\mathcal{P}_{XY})-1) = y_{J_m}$; combined with the above, this implies $\mathrm{sign}(\sigma_{m,q}) = y_{J_m}$. Furthermore, Lemma 29 implies that on

$\bigcap_{j=0}^{4} E_j$, $y_{J_m} = f^\star_{\mathcal{P}_{XY}}(X^2_m)$, so that $\mathrm{sign}(\sigma_{m,q}) = f^\star_{\mathcal{P}_{XY}}(X^2_m)$. In particular, recall that if $\hat{y}_{\infty,m} \in \mathcal{Y}$, then $|\sigma_{m,\hat{q}_{\infty,m}}| \geq 3\sqrt{2\hat{q}_{\infty,m} \ln\left(\frac{32\tilde{m}q_{\varepsilon,\delta}}{\delta}\right)}$. Thus, since the condition in Step 6 implies $\hat{q}_{\infty,m} \leq \tilde{q}_m \leq q_{\varepsilon,\delta}$, we have that on $\bigcap_{j=0}^{4} E_j$, if $\hat{y}_{\infty,m} \in \mathcal{Y}$, then $\hat{y}_{\infty,m} = f^\star_{\mathcal{P}_{XY}}(X^2_m)$. This completes the proof that $\hat{y}_{\infty,m} \in \{0, f^\star_{\mathcal{P}_{XY}}(X^2_m)\}$ on $\bigcap_{j=0}^{4} E_j$. Since we established above that $\hat{y}_{\infty,m} \in \mathcal{Y}$ if $\gamma_{J_m} \geq 2^{-k}$ on E_4, this also completes the proof that $\hat{y}_{\infty,m} = f^\star_{\mathcal{P}_{XY}}(X^2_m)$ when $\gamma_{J_m} \geq 2^{-k}$ on $\bigcap_{j=0}^{4} E_j$. ■

Lemma 36 *There exists an $(\mathbb{X}_1, \mathbb{X}_2)$-measurable event E_5 of probability at least $1 - \delta/64$ such that, on E_5, for every $k \in \{2, \ldots, k_\varepsilon\}$ with $\mathcal{P}\left(\bigcup\left\{A \in J : \gamma_A \in \left[2^{-k}, 2^{1-k}\right]\right\}\right) \geq 2^{k-3}\varepsilon/k_\varepsilon$,*

$$\left|\left\{m \in \{1, \ldots, \tilde{m}_k\} : \gamma_{J_m} \in \left[2^{-k}, 2^{1-k}\right]\right\}\right| \geq (1/2)\tilde{m}_k \mathcal{P}\left(\bigcup\left\{A \in J : \gamma_A \in \left[2^{-k}, 2^{1-k}\right]\right\}\right).$$

Proof Fix any $k \in \{2, \ldots, k_\varepsilon\}$. First, note that a Chernoff bound (under the conditional distribution given J) implies that, with conditional probability (given J) at least

$$1 - \exp\left\{-\frac{\tilde{m}_k}{8}\mathcal{P}\left(\bigcup\left\{A \in J : \gamma_A \in \left[2^{-k}, 2^{1-k}\right]\right\}\right)\right\},$$

we have

$$\left|\left\{m \in \{1, \ldots, \tilde{m}_k\} : \gamma_{J_m} \in \left[2^{-k}, 2^{1-k}\right]\right\}\right| \geq \frac{\tilde{m}_k}{2}\mathcal{P}\left(\bigcup\left\{A \in J : \gamma_A \in \left[2^{-k}, 2^{1-k}\right]\right\}\right). \quad (24)$$

If $\mathcal{P}\left(\bigcup\left\{A \in J : \gamma_A \in \left[2^{-k}, 2^{1-k}\right]\right\}\right) \geq 2^{k-3}\varepsilon/k_\varepsilon$, then

$$\exp\left\{-\frac{\tilde{m}_k}{8}\mathcal{P}\left(\bigcup\left\{A \in J : \gamma_A \in \left[2^{-k}, 2^{1-k}\right]\right\}\right)\right\}$$
$$\leq \exp\left\{-\frac{8k_\varepsilon}{2^k\varepsilon}\mathrm{Log}\left(\frac{64k_\varepsilon}{\delta}\right)2^{k-3}\varepsilon/k_\varepsilon\right\} = \exp\left\{-\mathrm{Log}\left(\frac{64k_\varepsilon}{\delta}\right)\right\} = \frac{\delta}{64k_\varepsilon}.$$

Thus, by the law of total probability, there is an event $G_5(k)$ of probability at least $1 - \delta/(64k_\varepsilon)$ such that, on $G_5(k)$, if $\mathcal{P}\left(\bigcup\left\{A \in J : \gamma_A \in \left[2^{-k}, 2^{1-k}\right]\right\}\right) \geq 2^{k-3}\varepsilon/k_\varepsilon$, then (24) holds. This holds for all $k \in \{2, \ldots, k_\varepsilon\}$ on the event $E_5 = \bigcap_{k=2}^{k_\varepsilon} G_5(k)$, which has probability at least $1 - \delta/64$ by a union bound. ■

We are now ready to apply the above results to characterize the behavior of Algorithm 1. For simplicity, we begin with the case of an infinite budget n, so that the algorithm proceeds until $m = \tilde{m}$; later, we discuss sufficient finite sizes of n to retain this behavior.

Lemma 37 *Consider running Algorithm 1 with budget ∞. On the event $\bigcap_{j=0}^{4} E_j$, $\forall k \in \{2, \ldots, k_\varepsilon\}$, $\forall m \in \{1, \ldots, \tilde{m}_k\}$, $f^\star_{\mathcal{P}_{XY}} \in V_m$ and*

$$V_m \subseteq \left\{h \in \mathbb{C} : \forall m' \leq m \text{ with } \gamma_{J_{m'}} \geq 2^{-k}, h(X^2_{m'}) = f^\star_{\mathcal{P}_{XY}}(X^2_{m'})\right\}.$$

Proof Fix any $k \in \{2, \ldots, k_\varepsilon\}$. We proceed by induction. The claim is clearly satisfied for $V_0 = \mathbb{C}$. Now take as the inductive hypothesis that, for some $m \in \{1, \ldots, \tilde{m}_k\}$, $f^\star_{\mathcal{P}_{XY}} \in V_{m-1} \subseteq \left\{h \in \mathbb{C} : \forall m' \leq m-1 \text{ with } \gamma_{J_{m'}} \geq 2^{-k}, h(X^2_{m'}) = f^\star_{\mathcal{P}_{XY}}(X^2_{m'})\right\}$.

If $X^2_m \notin \mathrm{DIS}(V_{m-1})$, then we have $V_m = V_{m-1}$, so that $f^\star_{\mathcal{P}_{XY}} \in V_m$ as well. Furthermore, since $f^\star_{\mathcal{P}_{XY}} \in V_{m-1}$, the fact that $X^2_m \notin \mathrm{DIS}(V_{m-1})$ implies that every $h \in V_m$ has $h(X^2_m) = f^\star_{\mathcal{P}_{XY}}(X^2_m)$. Therefore,

$$\begin{aligned}
V_m &= V_{m-1} \cap \left\{h \in \mathbb{C} : h(X^2_m) = f^\star_{\mathcal{P}_{XY}}(X^2_m)\right\} \\
&\subseteq \left\{h \in \mathbb{C} : \forall m' \leq m-1 \text{ with } \gamma_{J_{m'}} \geq 2^{-k}, h(X^2_{m'}) = f^\star_{\mathcal{P}_{XY}}(X^2_{m'})\right\} \\
&\qquad\qquad \cap \left\{h \in \mathbb{C} : h(X^2_m) = f^\star_{\mathcal{P}_{XY}}(X^2_m)\right\} \\
&\subseteq \left\{h \in \mathbb{C} : \forall m' \leq m \text{ with } \gamma_{J_{m'}} \geq 2^{-k}, h(X^2_{m'}) = f^\star_{\mathcal{P}_{XY}}(X^2_{m'})\right\}.
\end{aligned}$$

Next, consider the case that $X^2_m \in \mathrm{DIS}(V_{m-1})$. Lemma 35 implies that on $\bigcap_{j=0}^4 E_j$, $\hat{y}_{\infty,m} \in \{0, f^\star_{\mathcal{P}_{XY}}(X^2_m)\}$. If $\hat{y}_{\infty,m} = 0$, then $V_m = V_{m-1}$, so that $f^\star_{\mathcal{P}_{XY}} \in V_m$ by the inductive hypothesis. Furthermore, since $k \leq \tilde{k}_m$, Lemma 35 implies that on $\bigcap_{j=0}^4 E_j$, if $\gamma_{J_m} \geq 2^{-k}$ then $\hat{y}_{\infty,m} \neq 0$; thus, if $\hat{y}_{\infty,m} = 0$, we have $\gamma_{J_m} < 2^{-k}$, so that

$$\begin{aligned}
V_m = V_{m-1} &\subseteq \left\{h \in \mathbb{C} : \forall m' \leq m-1 \text{ with } \gamma_{J_{m'}} \geq 2^{-k}, h(X^2_{m'}) = f^\star_{\mathcal{P}_{XY}}(X^2_{m'})\right\} \\
&= \left\{h \in \mathbb{C} : \forall m' \leq m \text{ with } \gamma_{J_{m'}} \geq 2^{-k}, h(X^2_{m'}) = f^\star_{\mathcal{P}_{XY}}(X^2_{m'})\right\}.
\end{aligned}$$

On the other hand, if $\hat{y}_{\infty,m} = f^\star_{\mathcal{P}_{XY}}(X^2_m)$, then since $f^\star_{\mathcal{P}_{XY}} \in V_{m-1}$ by the inductive hypothesis, the condition in Step 5 will be satisfied, so that we have $V_m = \left\{h \in V_{m-1} : h(X^2_m) = f^\star_{\mathcal{P}_{XY}}(X^2_m)\right\}$. In particular, this implies $f^\star_{\mathcal{P}_{XY}} \in V_m$ as well, and combined with the inductive hypothesis, we have

$$\begin{aligned}
V_m &= V_{m-1} \cap \left\{h \in \mathbb{C} : h(X^2_m) = f^\star_{\mathcal{P}_{XY}}(X^2_m)\right\} \\
&\subseteq \left\{h \in \mathbb{C} : \forall m' \leq m \text{ with } \gamma_{J_{m'}} \geq 2^{-k}, h(X^2_{m'}) = f^\star_{\mathcal{P}_{XY}}(X^2_{m'})\right\}.
\end{aligned}$$

The result follows by the principle of induction. ∎

In particular, this implies the following result.

Lemma 38 *There exists an event E_6 of probability at least $1 - \delta/64$ such that, on $\bigcap_{j=0}^6 E_j$, the classifier $\hat{h}_\infty$ produced by Algorithm 1 with budget ∞ has $\mathrm{er}_{\mathcal{P}_{XY}}(\hat{h}_\infty) - \mathrm{er}_{\mathcal{P}_{XY}}(f^\star_{\mathcal{P}_{XY}}) \leq \varepsilon$.*

Proof Fix any $k \in \{2, \ldots, k_\varepsilon\}$ and let $\hat{\ell}_k = \left\lceil (1/2)\tilde{m}_k \mathcal{P}\left(\bigcup\left\{A \in J : \gamma_A \in \left[2^{-k}, 2^{1-k}\right]\right\}\right)\right\rceil$. Note that

$$\hat{\ell}_k \geq \frac{8ck_\varepsilon \mathcal{P}\left(\bigcup\left\{A \in J : \gamma_A \in \left[2^{-k}, 2^{1-k}\right]\right\}\right)}{2^k \varepsilon}\left(d\mathrm{Log}\left(\frac{8k_\varepsilon \mathcal{P}\left(\bigcup\left\{A \in J : \gamma_A \in \left[2^{-k}, 2^{1-k}\right]\right\}\right)}{2^k \varepsilon}\right) + \mathrm{Log}\left(\frac{64k_\varepsilon}{\delta}\right)\right),$$

for c as in Lemma 21. Let $\hat{m}_k = \min\left\{m \in \mathbb{N} : \sum_{m'=1}^{m} \mathbb{1}_{\left[2^{-k},2^{1-k}\right]}(\gamma_{J_{m'}}) = \hat{\ell}_k\right\} \cup \{\infty\}$. Note that, if $\hat{m}_k < \infty$, then the sequence $\left\{X_m^2 : 1 \le m \le \hat{m}_k, \gamma_{J_m} \in \left[2^{-k}, 2^{1-k}\right]\right\}$ is conditionally i.i.d. (given J and $\hat{m}_k$), with conditional distributions $\mathcal{P}\left(\cdot \middle| \bigcup\left\{A \in J : \gamma_A \in \left[2^{-k}, 2^{1-k}\right]\right\}\right)$. Applying Lemma 21 to these samples implies that there exists an event of conditional probability (given J and $\hat{m}_k$) at least $1 - \delta/(64k_\varepsilon)$ on which, if we have $\hat{m}_k < \infty$ and $\mathcal{P}\left(\bigcup\left\{A \in J : \gamma_A \in \left[2^{-k}, 2^{1-k}\right]\right\}\right) > \frac{2^k\varepsilon}{8k_\varepsilon}$, then letting

$$\mathcal{H}_k = \left\{h \in \mathbb{C} : \forall m \le \hat{m}_k \text{ with } \gamma_{J_m} \in \left[2^{-k}, 2^{1-k}\right], h(X_m^2) = f^\star_{\mathcal{P}_{XY}}(X_m^2)\right\},$$

every $h \in \mathcal{H}_k$ has

$$\mathcal{P}\left(x : h(x) \ne f^\star_{\mathcal{P}_{XY}}(x) \middle| \gamma_{J(x)} \in \left[2^{-k}, 2^{1-k}\right]\right) \le \frac{2^k\varepsilon}{8k_\varepsilon \mathcal{P}\left(\bigcup\{A \in J : \gamma_A \in [2^{-k}, 2^{1-k}]\}\right)},$$

which implies

$$\mathcal{P}\left(x : h(x) \ne f^\star_{\mathcal{P}_{XY}}(x) \text{ and } \gamma_{J(x)} \in \left[2^{-k}, 2^{1-k}\right]\right) \le \frac{2^k\varepsilon}{8k_\varepsilon}.$$

By the law of total probability and a union bound, there exists an event E_6 of probability at least $1 - \delta/64$ on which this holds for every $k \in \{2, \ldots, k_\varepsilon\}$.

Lemma 37 implies that, on $\bigcap_{j=0}^{4} E_j$, $\forall k \in \{2, \ldots, k_\varepsilon\}$,

$$V_{\tilde{m}} \subseteq V_{\tilde{m}_k} \subseteq \left\{h \in \mathbb{C} : \forall m \le \tilde{m}_k \text{ with } \gamma_{J_m} \ge 2^{-k}, h(X_m^2) = f^\star_{\mathcal{P}_{XY}}(X_m^2)\right\}.$$

Lemma 36 implies that, on E_5, $\forall k \in \{2, \ldots, k_\varepsilon\}$, if $\mathcal{P}\left(\bigcup\left\{A \in J : \gamma_A \in \left[2^{-k}, 2^{1-k}\right]\right\}\right) > \frac{2^k\varepsilon}{8k_\varepsilon}$, then $\left|\left\{m \in \{1, \ldots, \tilde{m}_k\} : \gamma_{J_m} \in \left[2^{-k}, 2^{1-k}\right]\right\}\right| \ge (1/2)\tilde{m}_k \mathcal{P}\left(\bigcup\left\{A \in J : \gamma_A \in \left[2^{-k}, 2^{1-k}\right]\right\}\right)$, so that $\hat{m}_k \le \tilde{m}_k$. In particular, this implies $\hat{m}_k < \infty$ and

$$\left\{h \in \mathbb{C} : \forall m \le \tilde{m}_k \text{ with } \gamma_{J_m} \ge 2^{-k}, h(X_m^2) = f^\star_{\mathcal{P}_{XY}}(X_m^2)\right\} \subseteq \mathcal{H}_k.$$

Combining the above three results, we have that on $\bigcap_{j=0}^{6} E_j$, for every $k \in \{2, \ldots, k_\varepsilon\}$ with $\mathcal{P}\left(\bigcup\left\{A \in J : \gamma_A \in \left[2^{-k}, 2^{1-k}\right]\right\}\right) > \frac{2^k\varepsilon}{8k_\varepsilon}$, $V_{\tilde{m}} \subseteq \mathcal{H}_k$, and therefore every $h \in V_{\tilde{m}}$ has

$$\mathcal{P}\left(x : h(x) \ne f^\star_{\mathcal{P}_{XY}}(x) \text{ and } \gamma_{J(x)} \in \left[2^{-k}, 2^{1-k}\right]\right) \le \frac{2^k\varepsilon}{8k_\varepsilon}.$$

Furthermore, for every $k \in \{2, \ldots, k_\varepsilon\}$ with $\mathcal{P}\left(\bigcup\left\{A \in J : \gamma_A \in \left[2^{-k}, 2^{1-k}\right]\right\}\right) \le \frac{2^k\varepsilon}{8k_\varepsilon}$, we also have that every $h \in V_{\tilde{m}}$ satisfies

$$\mathcal{P}\left(x : h(x) \ne f^\star_{\mathcal{P}_{XY}}(x) \text{ and } \gamma_{J(x)} \in \left[2^{-k}, 2^{1-k}\right]\right) \le \mathcal{P}\left(\bigcup\left\{A \in J : \gamma_A \in \left[2^{-k}, 2^{1-k}\right]\right\}\right) \le \frac{2^k\varepsilon}{8k_\varepsilon}.$$

Combined with Lemma 34, we have that on $\bigcap_{j=0}^{6} E_j$, every $h \in V_{\tilde{m}}$ has

$$\begin{aligned}
&\mathrm{er}_{\mathcal{P}_{XY}}(h) - \mathrm{er}_{\mathcal{P}_{XY}}(f^\star_{\mathcal{P}_{XY}}) \le 5\tau + \int \mathbb{1}\left[h(x) \neq f^\star_{\mathcal{P}_{XY}}(x)\right] 2\gamma_{J(x)} \mathcal{P}(\mathrm{d}x) \\
&\le 5\tau + 2^{1-k_\varepsilon} \mathcal{P}\left(x : h(x) \neq f^\star_{\mathcal{P}_{XY}}(x) \text{ and } \gamma_{J(x)} \le 2^{-k_\varepsilon}\right) \\
&\quad + \sum_{k=2}^{k_\varepsilon} 2^{2-k} \mathcal{P}\left(x : h(x) \neq f^\star_{\mathcal{P}_{XY}}(x) \text{ and } \gamma_{J(x)} \in \left[2^{-k}, 2^{1-k}\right]\right) \\
&\le 5\tau + 2^{1-k_\varepsilon} \mathcal{P}\left(\bigcup \left\{A \in J : \gamma_A \le 2^{-k_\varepsilon}\right\}\right) + \sum_{k=2}^{k_\varepsilon} 2^{2-k} \frac{2^k \varepsilon}{8 k_\varepsilon}. \qquad (25)
\end{aligned}$$

Next, note that $\sum_{k=2}^{k_\varepsilon} 2^{2-k} \frac{2^k \varepsilon}{8k_\varepsilon} = (k_\varepsilon - 1)\frac{\varepsilon}{2k_\varepsilon} \le \frac{\varepsilon}{2}$. Furthermore, since $2^{-k_\varepsilon} \le \hat{\gamma}_\varepsilon/8 < \gamma_\varepsilon/4$, Lemma 33 implies that, on E_1,

$$\mathcal{P}\left(\bigcup \left\{A \in J : \gamma_A \le 2^{-k_\varepsilon}\right\}\right) \le \frac{3\varepsilon}{2\gamma_\varepsilon}.$$

Plugging these facts into (25) reveals that, on $\bigcap_{j=0}^{6} E_j$, $\forall h \in V_{\tilde{m}}$,

$$\mathrm{er}_{\mathcal{P}_{XY}}(h) - \mathrm{er}_{\mathcal{P}_{XY}}(f^\star_{\mathcal{P}_{XY}}) \le 5\tau + 2^{1-k_\varepsilon} \frac{3\varepsilon}{2\gamma_\varepsilon} + \frac{\varepsilon}{2} \le 5\tau + \frac{3}{8}\varepsilon + \frac{\varepsilon}{2} \le \frac{453}{512}\varepsilon < \varepsilon.$$

The result follows by noting that, when the budget is set to ∞, Algorithm 1 definitely reaches $m = \tilde{m}$ before halting, so that $\hat{h}_\infty \in V_{\tilde{m}}$. ■

The only remaining question is how many label requests the algorithm makes in the process of producing this $\hat{h}_\infty$, so that taking a budget n of at least this size is equivalent to having an infinite budget. This question is addressed by the following sequence of lemmas.

Lemma 39 *Consider running Algorithm 1 with budget ∞. There exists an event E_7 of probability at least $1 - \delta/64$ such that, on $E_1 \cap E_7$, $\forall k \in \{2, \ldots, k_\varepsilon\}$,*

$$\begin{aligned}
&\left|\left\{m \in \{1, \ldots, \tilde{m}_k\} : \gamma_{J_m} \le 2^{1-k}, X_m^2 \in \mathrm{DIS}(V_{m-1})\right\}\right| \\
&\qquad \le 17 \max\left\{\mathcal{P}\left(x : \gamma_x < 2^{3-k}\right), \frac{\varepsilon}{2\hat{\gamma}_\varepsilon}\right\} \tilde{m}_k.
\end{aligned}$$

Proof Fix any $k \in \{2, \ldots, k_\varepsilon\}$. By a Chernoff bound (applied under the conditional given J) and the law of total probability, there is an event $G_7(k)$ of probability at least $1 - \frac{\delta}{64k_\varepsilon}$, on which

$$\left|\left\{m \in \{1, \ldots, \tilde{m}_k\} : \gamma_{J_m} \le 2^{1-k}\right\}\right| \le \log_2\left(\frac{64k_\varepsilon}{\delta}\right) + 2e\mathcal{P}\left(\bigcup \left\{A \in J : \gamma_A \le 2^{1-k}\right\}\right) \tilde{m}_k.$$

Lemma 33 implies that, on E_1,

$$\mathcal{P}\left(\bigcup \left\{A \in J : \gamma_A \le 2^{1-k}\right\}\right) \le \max\left\{3\mathcal{P}\left(x : \gamma_x < 2^{3-k}\right), \frac{3\varepsilon}{2\gamma_\varepsilon}\right\}.$$

Therefore, on $E_1 \cap G_7(k)$,

$$\left|\left\{m \in \{1,\ldots,\tilde{m}_k\} : \gamma_{J_m} \le 2^{1-k}, X_m^2 \in \mathrm{DIS}(V_{m-1})\right\}\right| \le \left|\left\{m \in \{1,\ldots,\tilde{m}_k\} : \gamma_{J_m} \le 2^{1-k}\right\}\right|$$
$$\le \log_2\left(\frac{64k_\varepsilon}{\delta}\right) + 6e\max\left\{\mathcal{P}\left(x : \gamma_x < 2^{3-k}\right), \frac{\varepsilon}{2\gamma_\varepsilon}\right\}\tilde{m}_k. \tag{26}$$

Furthermore, since $\hat{\gamma}_\varepsilon \le \gamma_\varepsilon$, and

$$\frac{\varepsilon}{2\hat{\gamma}_\varepsilon}\tilde{m}_k \ge \frac{64}{2^{k_\varepsilon}\hat{\gamma}_\varepsilon}\mathrm{Log}\left(\frac{64k_\varepsilon}{\delta}\right) \ge 4\mathrm{Log}\left(\frac{64k_\varepsilon}{\delta}\right) \ge 2\log_2\left(\frac{64k_\varepsilon}{\delta}\right),$$

(26) is at most

$$\left(6e + \frac{1}{2}\right)\max\left\{\mathcal{P}\left(x : \gamma_x < 2^{3-k}\right), \frac{\varepsilon}{2\hat{\gamma}_\varepsilon}\right\}\tilde{m}_k \le 17\max\left\{\mathcal{P}\left(x : \gamma_x < 2^{3-k}\right), \frac{\varepsilon}{2\hat{\gamma}_\varepsilon}\right\}\tilde{m}_k.$$

Defining $E_7 = \bigcap_{k=2}^{k_\varepsilon} G_7(k)$, a union bound implies E_7 has probability at least $1-\delta/64$, and the result follows. ∎

Lemma 40 *Consider running Algorithm 1 with budget ∞. There exists an event E_8 of probability at least $1-\delta/64$ such that, on $E_8 \cap \bigcap_{j=0}^4 E_j$, $\forall \bar{k} \in \{3,\ldots,k_\varepsilon\}$, $\forall k \in \{2,\ldots,\bar{k}-1\}$,*

$$\left|\left\{m \in \{1,\ldots,\tilde{m}_k\} : X_m^2 \in \mathrm{DIS}(V_{m-1})\right\}\right|$$
$$\le 6e\max\left\{\mathcal{P}\left(x : \gamma_x < 2^{2-\bar{k}}\right), \frac{\varepsilon}{2\gamma_\varepsilon}\right\}\tilde{m}_k$$
$$+ 91\tilde{c}\left(2^{1+\bar{k}-k} + \mathrm{Log}\left(\frac{64c}{\varepsilon}\right)\right)\left(6\mathfrak{s}\mathrm{Log}\left(\frac{128c}{\varepsilon}\right) + \mathrm{Log}\left(\frac{1}{\delta}\right)\right),$$

for c as in Lemma 21 and $\tilde{c}$ as in Lemma 23.

Proof The claim trivially holds if $\mathfrak{s} = \infty$, so for the remainder of the proof we suppose $\mathfrak{s} < \infty$. Fix any $\bar{k} \in \{3,\ldots,k_\varepsilon\}$ and $k \in \{2,\ldots,\bar{k}-1\}$, and note that

$$\left|\left\{m \in \{1,\ldots,\tilde{m}_k\} : X_m^2 \in \mathrm{DIS}(V_{m-1})\right\}\right|$$
$$\le \left|\left\{m \in \{1,\ldots,\tilde{m}_k\} : \gamma_{J_m} \le 2^{-\bar{k}}, X_m^2 \in \mathrm{DIS}(V_{m-1})\right\}\right|$$
$$+ \left|\left\{m \in \{1,\ldots,\tilde{m}_k\} : \gamma_{J_m} \ge 2^{-\bar{k}}, X_m^2 \in \mathrm{DIS}(V_{m-1})\right\}\right|. \tag{27}$$

We proceed to bound each term on the right hand side. A Chernoff bound (applied under the conditional distribution given J) and the law of total probability imply that, on an event $G_8^{(i)}(\bar{k},k)$ of probability at least $1 - \frac{\delta}{256k_\varepsilon^2}$,

$$\left|\left\{m \in \{1,\ldots,\tilde{m}_k\} : \gamma_{J_m} \le 2^{-\bar{k}}, X_m^2 \in \mathrm{DIS}(V_{m-1})\right\}\right|$$
$$\le \log_2\left(\frac{256k_\varepsilon^2}{\delta}\right) + 2e\mathcal{P}\left(\bigcup\left\{A \in J : \gamma_A \le 2^{-\bar{k}}\right\}\right)\tilde{m}_k,$$

and Lemma 33 implies that, on E_1, this is at most

$$\log_2\left(\frac{256k_\varepsilon^2}{\delta}\right) + 6e\max\left\{\mathcal{P}\left(x : \gamma_x < 2^{2-\bar{k}}\right), \frac{\varepsilon}{2\gamma_\varepsilon}\right\}\tilde{m}_k.$$

Now we turn to bounding the second term on the right hand side of (27). We proceed in two steps, noting that monotonicity of $m \mapsto \mathrm{DIS}(V_m)$ implies

$$\begin{aligned}\left|\left\{m \in \{1,\ldots,\tilde{m}_k\} : \gamma_{J_m} \geq 2^{-\bar{k}}, X_m^2 \in \mathrm{DIS}(V_{m-1})\right\}\right| \\ \leq \left|\left\{m \in \{1,\ldots,\tilde{m}_{\bar{k}}\} : \gamma_{J_m} \geq 2^{-\bar{k}}, X_m^2 \in \mathrm{DIS}(V_{m-1})\right\}\right| \\ + \left|\left\{m \in \{\tilde{m}_{\bar{k}}+1,\ldots,\tilde{m}_k\} : \gamma_{J_m} \geq 2^{-\bar{k}}, X_m^2 \in \mathrm{DIS}(V_{\tilde{m}_{\bar{k}}})\right\}\right|. \end{aligned} \tag{28}$$

We start with the first term on the right of (28). Let $L = \left|\left\{m \in \{1,\ldots,\tilde{m}_{\bar{k}}\} : \gamma_{J_m} \geq 2^{-\bar{k}}\right\}\right|$, and let $\ell_1,\ldots,\ell_L$ denote the increasing subsequence of values $\ell \in \{1,\ldots,\tilde{m}_{\bar{k}}\}$ with $\gamma_{J_\ell} \geq 2^{-\bar{k}}$. Also, let $\tilde{j}_{\bar{k}} = \max\{1, \lceil \log_2(\tilde{m}_{\bar{k}}/(\mathfrak{s} + \mathrm{Log}(1/\delta)))\rceil\}$, let $M_0 = 0$, and for each $j \in \mathbb{N}$, let

$$M_j = \left\lceil \tilde{c}2^j\left(\mathfrak{s}\mathrm{Log}\left(2^j\right) + \mathrm{Log}\left(\frac{256k_\varepsilon^2\tilde{j}_{\bar{k}}}{\delta}\right)\right)\right\rceil,$$

for $\tilde{c}$ as in Lemma 23. Let $V_0^\star = \mathbb{C}$, and for each $i \leq L$, let

$$V_i^\star = \left\{h \in \mathbb{C} : \forall j \in \{1,\ldots,i\}, h(X_{\ell_j}^2) = f^\star_{\mathcal{P}_{XY}}(X_{\ell_j}^2)\right\}.$$

Let $\phi_{\mathfrak{s}}$ be the function mapping any $\mathcal{U} \in \mathcal{X}^{\mathfrak{s}}$ to the set $\mathrm{DIS}(\{h \in \mathbb{C} : \forall x \in \mathcal{U}, h(x) = f^\star_{\mathcal{P}_{XY}}(x)\})$. Fix any $j \in \mathbb{N}$. By Theorem 13, if $M_j \leq L$, then there exist $i_1,\ldots,i_{\mathfrak{s}} \in \{1,\ldots,M_j\}$ such that $\{h \in \mathbb{C} : \forall r \in \{1,\ldots,\mathfrak{s}\}, h(X_{\ell_{i_r}}^2) = f^\star_{\mathcal{P}_{XY}}(X_{\ell_{i_r}}^2)\} = V^\star_{M_j}$ (see the discussion in Section 7.3.1). In particular, for this choice of $i_1,\ldots,i_{\mathfrak{s}}$, we have $\phi_{\mathfrak{s}}(X_{\ell_{i_1}}^2,\ldots,X_{\ell_{i_{\mathfrak{s}}}}^2) = \mathrm{DIS}(V^\star_{M_j})$; furthermore, since $\phi_{\mathfrak{s}}$ is permutation-invariant, we can take $i_1 \leq \cdots \leq i_{\mathfrak{s}}$ without loss of generality. Also note that $X_{\ell_1}^2,\ldots,X_{\ell_{M_j\wedge L}}^2$ are conditionally independent (given L and J), each with conditional distribution $\mathcal{P}\left(\cdot\middle|\bigcup\left\{A \in J : \gamma_A \geq 2^{-\bar{k}}\right\}\right)$. Since (when $M_j \leq L$) $\{X_{\ell_1}^2,\ldots,X_{\ell_{M_j}}^2\} \cap \phi_{\mathfrak{s}}(X_{\ell_{i_1}}^2,\ldots,X_{\ell_{i_{\mathfrak{s}}}}^2) = \{X_{\ell_1}^2,\ldots,X_{\ell_{M_j}}^2\} \cap \mathrm{DIS}(V^\star_{M_j}) = \emptyset$, Lemma 23 (applied under the conditional distribution given L and J) and the law of total probability imply that, on an event $G_8^{(ii)}(\bar{k},k,j)$ of probability at least $1 - \frac{\delta}{256k_\varepsilon^2\tilde{j}_{\bar{k}}}$, if $M_j \leq L$, then

$$\mathcal{P}\left(\mathrm{DIS}(V^\star_{M_j})\middle|\bigcup\left\{A \in J : \gamma_A \geq 2^{-\bar{k}}\right\}\right) \leq 2^{-j}. \tag{29}$$

Furthermore, this clearly holds for $j = 0$ as well. Since $\mathcal{P}\left(\mathrm{DIS}\left(V_{i-1}^\star\right)\middle|\bigcup\left\{A \in J : \gamma_A \geq 2^{-\bar{k}}\right\}\right)$ is nonincreasing in i, for every $j \geq 0$ with $M_j < L$, and every $i \in \{M_j+1,\ldots,M_{j+1}\wedge L\}$, on $G_8^{(ii)}(\bar{k},k,j)$, $\mathcal{P}\left(\mathrm{DIS}\left(V_{i-1}^\star\right)\middle|\bigcup\left\{A \in J : \gamma_A \geq 2^{-\bar{k}}\right\}\right) \leq 2^{-j}$. Since every $j \geq \tilde{j}_{\bar{k}}$ has $M_j \geq \tilde{m}_{\bar{k}} \geq L$, this holds simultaneously for every j with $M_j < L$ on $\bigcap_{j=1}^{\tilde{j}_{\bar{k}}-1} G_8^{(ii)}(\bar{k},k,j)$.

Now note that, conditioned on J and L,

$$\left\{\mathbb{1}_{\mathrm{DIS}(V_{i-1}^\star)}\left(X_{\ell_i}^2\right) - \mathcal{P}\left(\mathrm{DIS}\left(V_{i-1}^\star\right)\middle|\bigcup\left\{A \in J : \gamma_A \geq 2^{-\bar{k}}\right\}\right)\right\}_{i=1}^L$$

is a martingale difference sequence with respect to $X^2_{\ell_1}, \ldots, X^2_{\ell_L}$. Therefore, Bernstein's inequality for martingales (e.g., McDiarmid, 1998, Theorem 3.12), applied under the conditional distribution given J and L, along with the law of total probability, imply that there exists an event $G_8^{(iii)}(\bar{k}, k)$ of probability at least $1 - \frac{\delta}{256k_\varepsilon^2}$ such that, on $G_8^{(iii)}(\bar{k}, k) \cap \bigcap_{j=1}^{\tilde{j}_{\bar{k}}-1} G_8^{(ii)}(\bar{k}, k, j)$,

$$\begin{aligned}\sum_{i=1}^{L} \mathbb{1}_{\mathrm{DIS}(V^\star_{i-1})}\left(X^2_{\ell_i}\right) &\le \log_2\left(\frac{256k_\varepsilon^2}{\delta}\right) + 2e\sum_{j=0}^{\tilde{j}_{\bar{k}}-1} 2^{-j}(M_{j+1} - M_j) \\ &\le \log_2\left(\frac{256k_\varepsilon^2}{\delta}\right) + 4e + 4e\tilde{c}\left(\mathfrak{s}\mathrm{Log}\left(2^{\tilde{j}_{\bar{k}}}\right) + \mathrm{Log}\left(\frac{256k_\varepsilon^2\tilde{j}_{\bar{k}}}{\delta}\right)\right)\tilde{j}_{\bar{k}} \\ &\le 8e\tilde{c}\left(\mathfrak{s}\tilde{j}_{\bar{k}} + \mathrm{Log}\left(\frac{256k_\varepsilon^2}{\delta}\right)\right)\tilde{j}_{\bar{k}}.\end{aligned}$$

By Lemma 37, on $\bigcap_{j=0}^{4} E_j$, $\forall m \in \{1, \ldots, \tilde{m}_{\bar{k}}\}$,

$$V_m \subseteq \left\{h \in \mathbb{C} : \forall m' \le m \text{ with } \gamma_{J_{m'}} \ge 2^{-\bar{k}}, h(X^2_{m'}) = f^\star_{\mathcal{P}_{XY}}(X^2_{m'})\right\}.$$

In particular, this implies $V_{\ell_i - 1} \subseteq V^\star_{i-1}$ for all $i \le L$. Therefore, on $\bigcap_{j=0}^{4} E_j \cap G_8^{(iii)}(\bar{k}, k) \cap \bigcap_{j=1}^{\tilde{j}_{\bar{k}}-1} G_8^{(ii)}(\bar{k}, k, j)$,

$$\begin{aligned}\left|\left\{m \in \{1, \ldots, \tilde{m}_{\bar{k}}\} : \gamma_{J_m} \ge 2^{-\bar{k}}, X^2_m \in \mathrm{DIS}(V_{m-1})\right\}\right| &= \sum_{i=1}^{L} \mathbb{1}_{\mathrm{DIS}(V_{\ell_i - 1})}\left(X^2_{\ell_i}\right) \\ &\le \sum_{i=1}^{L} \mathbb{1}_{\mathrm{DIS}(V^\star_{i-1})}\left(X^2_{\ell_i}\right) \le 8e\tilde{c}\left(\mathfrak{s}\tilde{j}_{\bar{k}} + \mathrm{Log}\left(\frac{256k_\varepsilon^2}{\delta}\right)\right)\tilde{j}_{\bar{k}}. \quad (30)\end{aligned}$$

Next, we turn to bounding the second term on the right hand side of (28). A Chernoff bound (applied under the conditional distribution given $V_{\tilde{m}_{\bar{k}}}$ and J) and the law of total probability imply that there is an event $G_8^{(iv)}(\bar{k}, k)$ of probability at least $1 - \frac{\delta}{256k_\varepsilon^2}$, on which

$$\begin{aligned}&\left|\left\{m \in \{\tilde{m}_{\bar{k}} + 1, \ldots, \tilde{m}_k\} : \gamma_{J_m} \ge 2^{-\bar{k}}, X^2_m \in \mathrm{DIS}(V_{\tilde{m}_{\bar{k}}})\right\}\right| \\ &\qquad \le \log_2\left(\frac{256k_\varepsilon^2}{\delta}\right) + 2e\mathcal{P}\left(\mathrm{DIS}\left(V_{\tilde{m}_{\bar{k}}}\right) \cap \bigcup\left\{A \in J : \gamma_A \ge 2^{-\bar{k}}\right\}\right)\tilde{m}_k. \quad (31)\end{aligned}$$

Also, by a Chernoff bound (applied under the conditional distribution given J), with probability at least

$$1 - \exp\left\{-(1/8)\mathcal{P}\left(\bigcup\left\{A \in J : \gamma_A \ge 2^{-\bar{k}}\right\}\right)\tilde{m}_{\bar{k}}\right\},$$

we have

$$L \ge (1/2)\tilde{m}_{\bar{k}}\mathcal{P}\left(\bigcup\left\{A \in J : \gamma_A \ge 2^{-\bar{k}}\right\}\right). \quad (32)$$

If $\mathcal{P}\left(\bigcup\left\{A \in J : \gamma_A \geq 2^{-\bar{k}}\right\}\right) \geq \frac{8}{\tilde{m}_{\bar{k}}}\mathrm{Log}\left(\frac{256k_\varepsilon}{\delta}\right)$, then

$$\exp\left\{-(1/8)\mathcal{P}\left(\bigcup\left\{A \in J : \gamma_A \geq 2^{-\bar{k}}\right\}\right)\tilde{m}_{\bar{k}}\right\} \leq \frac{\delta}{256k_\varepsilon}.$$

Thus, by the law of total probability, there is an event $G_8^{(v)}(\bar{k})$ of probability at least $1-\frac{\delta}{256k_\varepsilon}$, on which, if $\mathcal{P}\left(\bigcup\left\{A \in J : \gamma_A \geq 2^{-\bar{k}}\right\}\right) \geq \frac{8}{\tilde{m}_{\bar{k}}}\mathrm{Log}\left(\frac{256k_\varepsilon}{\delta}\right)$, then (32) holds. Let

$$\hat{j} = \max\left\{j \in \{0,1,\ldots,\tilde{j}_{\bar{k}}-1\} : M_j \leq (1/2)\tilde{m}_{\bar{k}}\mathcal{P}\left(\bigcup\left\{A \in J : \gamma_A \geq 2^{-\bar{k}}\right\}\right)\right\},$$

and note that

$$\hat{j} \geq \left\lfloor \log_2\left(\frac{\tilde{m}_{\bar{k}}\mathcal{P}\left(\bigcup\left\{A \in J : \gamma_A \geq 2^{-\bar{k}}\right\}\right)}{4\tilde{c}\left(2\mathfrak{s}\mathrm{Log}\left(2^{\tilde{j}_{\bar{k}}}\right)+\mathrm{Log}\left(\frac{256k_\varepsilon^2}{\delta}\right)\right)}\right)\right\rfloor. \tag{33}$$

(29) implies that on $\bigcap_{j=1}^{\tilde{j}_{\bar{k}}-1} G_8^{(ii)}(\bar{k},k,j)$, if (32) holds, we have

$$\mathcal{P}\left(\mathrm{DIS}\left(V_L^\star\right)\middle|\bigcup\left\{A \in J : \gamma_A \geq 2^{-\bar{k}}\right\}\right) \leq 2^{-\hat{j}}.$$

Furthermore, Lemma 37 implies that, on $\bigcap_{j=0}^4 E_j$, $V_{\tilde{m}_{\bar{k}}} \subseteq V_L^\star$. Altogether, on $\bigcap_{j=0}^4 E_j \cap G_8^{(v)}(\bar{k}) \cap \bigcap_{j=1}^{\tilde{j}_{\bar{k}}-1} G_8^{(ii)}(\bar{k},k,j)$, if $\mathcal{P}\left(\bigcup\left\{A \in J : \gamma_A \geq 2^{-\bar{k}}\right\}\right) \geq \frac{8}{\tilde{m}_{\bar{k}}}\mathrm{Log}\left(\frac{256k_\varepsilon}{\delta}\right)$, then

$$\begin{aligned}\mathcal{P}\left(\mathrm{DIS}\left(V_{\tilde{m}_{\bar{k}}}\right)\cap\bigcup\left\{A \in J : \gamma_A \geq 2^{-\bar{k}}\right\}\right) &\leq 2^{-\hat{j}}\mathcal{P}\left(\bigcup\left\{A \in J : \gamma_A \geq 2^{-\bar{k}}\right\}\right)\\ &\leq \frac{8\tilde{c}}{\tilde{m}_{\bar{k}}}\left(2\mathfrak{s}\mathrm{Log}\left(2^{\tilde{j}_{\bar{k}}}\right)+\mathrm{Log}\left(\frac{256k_\varepsilon^2}{\delta}\right)\right),\end{aligned}$$

where the last inequality is by (33). Otherwise, if $\mathcal{P}\left(\bigcup\left\{A \in J : \gamma_A \geq 2^{-\bar{k}}\right\}\right) < \frac{8}{\tilde{m}_{\bar{k}}}\mathrm{Log}\left(\frac{256k_\varepsilon}{\delta}\right)$, then in any case we have

$$\begin{aligned}\mathcal{P}\left(\mathrm{DIS}\left(V_{\tilde{m}_{\bar{k}}}\right)\cap\bigcup\left\{A \in J : \gamma_A \geq 2^{-\bar{k}}\right\}\right) &\leq \mathcal{P}\left(\bigcup\left\{A \in J : \gamma_A \geq 2^{-\bar{k}}\right\}\right)\\ &< \frac{8}{\tilde{m}_{\bar{k}}}\mathrm{Log}\left(\frac{256k_\varepsilon}{\delta}\right) \leq \frac{8\tilde{c}}{\tilde{m}_{\bar{k}}}\left(2\mathfrak{s}\mathrm{Log}\left(2^{\tilde{j}_{\bar{k}}}\right)+\mathrm{Log}\left(\frac{256k_\varepsilon^2}{\delta}\right)\right).\end{aligned}$$

Combined with (31), this implies that on $\bigcap_{j=0}^4 E_j \cap G_8^{(iv)}(\bar{k},k) \cap G_8^{(v)}(\bar{k}) \cap \bigcap_{j=1}^{\tilde{j}_{\bar{k}}-1} G_8^{(ii)}(\bar{k},k,j)$,

$$\begin{aligned}&\left|\left\{m \in \{\tilde{m}_{\bar{k}}+1,\ldots,\tilde{m}_k\} : \gamma_{J_m} \geq 2^{-\bar{k}}, X_m^2 \in \mathrm{DIS}(V_{\tilde{m}_{\bar{k}}})\right\}\right|\\ &\leq \log_2\left(\frac{256k_\varepsilon^2}{\delta}\right)+16e\tilde{c}\frac{\tilde{m}_k}{\tilde{m}_{\bar{k}}}\left(2\mathfrak{s}\mathrm{Log}\left(2^{\tilde{j}_{\bar{k}}}\right)+\mathrm{Log}\left(\frac{256k_\varepsilon^2}{\delta}\right)\right)\\ &\leq 32e\tilde{c}\frac{\tilde{m}_k}{\tilde{m}_{\bar{k}}}\left(\mathfrak{s}\tilde{j}_{\bar{k}}+\mathrm{Log}\left(\frac{256k_\varepsilon^2}{\delta}\right)\right) \leq 64e\tilde{c}2^{\bar{k}-k}\left(\mathfrak{s}\tilde{j}_{\bar{k}}+\mathrm{Log}\left(\frac{256k_\varepsilon^2}{\delta}\right)\right).\end{aligned}$$

Plugging this and (30) into (28), we have that on $\bigcap_{j=0}^{4} E_j \cap G_8^{(iii)}(\bar{k},k) \cap G_8^{(iv)}(\bar{k},k) \cap G_8^{(v)}(\bar{k}) \cap \bigcap_{j=1}^{\tilde{j}_{\bar{k}}-1} G_8^{(ii)}(\bar{k},k,j)$,

$$\begin{aligned}
&\left|\left\{m \in \{1,\ldots,\tilde{m}_k\} : \gamma_{J_m} \geq 2^{-\bar{k}}, X_m^2 \in \mathrm{DIS}(V_{m-1})\right\}\right| \\
&\leq 8e\tilde{c}\left(\mathfrak{s}\tilde{j}_{\bar{k}} + \mathrm{Log}\left(\frac{256k_\varepsilon^2}{\delta}\right)\right)\tilde{j}_{\bar{k}} + 64e\tilde{c}2^{\bar{k}-k}\left(\mathfrak{s}\tilde{j}_{\bar{k}} + \mathrm{Log}\left(\frac{256k_\varepsilon^2}{\delta}\right)\right) \\
&= 8e\tilde{c}\left(2^{3+\bar{k}-k} + \tilde{j}_{\bar{k}}\right)\left(\mathfrak{s}\tilde{j}_{\bar{k}} + \mathrm{Log}\left(\frac{256k_\varepsilon^2}{\delta}\right)\right).
\end{aligned}$$

Combined with the above result bounding the first term in (27), we have that on $\bigcap_{j=0}^{4} E_j \cap G_8^{(i)}(\bar{k},k) \cap G_8^{(iii)}(\bar{k},k) \cap G_8^{(iv)}(\bar{k},k) \cap G_8^{(v)}(\bar{k}) \cap \bigcap_{j=1}^{\tilde{j}_{\bar{k}}-1} G_8^{(ii)}(\bar{k},k,j)$,

$$\begin{aligned}
&\left|\left\{m \in \{1,\ldots,\tilde{m}_k\} : X_m^2 \in \mathrm{DIS}(V_{m-1})\right\}\right| \\
&\leq \log_2\left(\frac{256k_\varepsilon^2}{\delta}\right) + 6e\max\left\{\mathcal{P}\left(x : \gamma_x < 2^{2-\bar{k}}\right), \frac{\varepsilon}{2\gamma_\varepsilon}\right\}\tilde{m}_k \\
&\qquad + 8e\tilde{c}\left(2^{3+\bar{k}-k} + \tilde{j}_{\bar{k}}\right)\left(\mathfrak{s}\tilde{j}_{\bar{k}} + \mathrm{Log}\left(\frac{256k_\varepsilon^2}{\delta}\right)\right) \\
&\leq 6e\max\left\{\mathcal{P}\left(x : \gamma_x < 2^{2-\bar{k}}\right), \frac{\varepsilon}{2\gamma_\varepsilon}\right\}\tilde{m}_k + (1+8e\tilde{c})\left(2^{3+\bar{k}-k} + \tilde{j}_{\bar{k}}\right)\left(\mathfrak{s}\tilde{j}_{\bar{k}} + \mathrm{Log}\left(\frac{256k_\varepsilon^2}{\delta}\right)\right).
\end{aligned} \tag{34}$$

Noting that $\mathfrak{s} \geq d$, a bit of algebra reveals that

$$\frac{\tilde{m}_{\bar{k}}}{\mathfrak{s} + \mathrm{Log}(1/\delta)} \leq \frac{32ck_\varepsilon}{\varepsilon}\mathrm{Log}\left(\frac{128k_\varepsilon^2}{\varepsilon}\right) \leq \frac{2^9 ck_\varepsilon^2}{\varepsilon^{3/2}},$$

so that

$$\tilde{j}_{\bar{k}} \leq \log_2\left(\frac{2^{10}ck_\varepsilon^2}{\varepsilon^{3/2}}\right) \leq \frac{3}{2}\mathrm{Log}\left(\frac{2^{10}ck_\varepsilon^2}{\varepsilon^{3/2}}\right),$$

and therefore

$$\begin{aligned}
&(1+8e\tilde{c})\left(2^{3+\bar{k}-k} + \tilde{j}_{\bar{k}}\right)\left(\mathfrak{s}\tilde{j}_{\bar{k}} + \mathrm{Log}\left(\frac{256k_\varepsilon^2}{\delta}\right)\right) \\
&\leq (1+8e\tilde{c})\left(2^{3+\bar{k}-k} + \frac{3}{2}\mathrm{Log}\left(\frac{2^{10}ck_\varepsilon^2}{\varepsilon^{3/2}}\right)\right)\left(\frac{3}{2}\mathfrak{s}\mathrm{Log}\left(\frac{2^{10}ck_\varepsilon^2}{\varepsilon^{3/2}}\right) + \mathrm{Log}\left(\frac{256k_\varepsilon^2}{\delta}\right)\right) \\
&\leq (1+8e\tilde{c})\left(2^{3+\bar{k}-k} + \frac{3}{2}\mathrm{Log}\left(\frac{2^{10}ck_\varepsilon^2}{\varepsilon^{3/2}}\right)\right)\left(\frac{3}{2}\mathfrak{s}\mathrm{Log}\left(\frac{2^{16}ck_\varepsilon^4}{\varepsilon^{3/2}}\right) + \mathrm{Log}\left(\frac{1}{\delta}\right)\right).
\end{aligned}$$

Furthermore, since $k_\varepsilon \leq \sqrt{32/\varepsilon}$, this is at most

$$\begin{aligned}
&(1+8e\tilde{c})\left(2^{3+\bar{k}-k} + \frac{3}{2}\mathrm{Log}\left(\frac{2^{15}c}{\varepsilon^{5/2}}\right)\right)\left(\frac{3}{2}\mathfrak{s}\mathrm{Log}\left(\frac{2^{26}c}{\varepsilon^{7/2}}\right) + \mathrm{Log}\left(\frac{1}{\delta}\right)\right) \\
&\leq 91\tilde{c}\left(2^{1+\bar{k}-k} + \mathrm{Log}\left(\frac{64c}{\varepsilon}\right)\right)\left(6\mathfrak{s}\mathrm{Log}\left(\frac{128c}{\varepsilon}\right) + \mathrm{Log}\left(\frac{1}{\delta}\right)\right).
\end{aligned}$$

Plugging this into (34), we have that on $\bigcap_{j=0}^{4} E_j \cap G_8^{(i)}(\bar{k},k) \cap G_8^{(iii)}(\bar{k},k) \cap G_8^{(iv)}(\bar{k},k) \cap G_8^{(v)}(\bar{k}) \cap \bigcap_{j=1}^{\tilde{j}_{\bar{k}}-1} G_8^{(ii)}(\bar{k},k,j)$,

$$\begin{aligned}\left|\left\{m \in \{1,\ldots,\tilde{m}_k\} : X_m^2 \in \mathrm{DIS}(V_{m-1})\right\}\right| &\\ \leq 6e \max\left\{\mathcal{P}\left(x : \gamma_x < 2^{2-\bar{k}}\right), \frac{\varepsilon}{2\gamma_\varepsilon}\right\} \tilde{m}_k &\\ + 91\tilde{c}\left(2^{1+\bar{k}-k} + \mathrm{Log}\left(\frac{64c}{\varepsilon}\right)\right)\left(6\mathfrak{s}\mathrm{Log}\left(\frac{128c}{\varepsilon}\right) + \mathrm{Log}\left(\frac{1}{\delta}\right)\right). \quad (35)\end{aligned}$$

Letting

$$E_8 = \bigcap_{\bar{k}=3}^{k_\varepsilon}\left(G_8^{(v)}(\bar{k}) \cap \bigcap_{k=2}^{\bar{k}-1} G_8^{(i)}(\bar{k},k) \cap G_8^{(iii)}(\bar{k},k) \cap G_8^{(iv)}(\bar{k},k) \cap \bigcap_{j=1}^{\tilde{j}_{\bar{k}}-1} G_8^{(ii)}(\bar{k},k,j)\right),$$

we have that (35) holds for all $\bar{k} \in \{3,\ldots,k_\varepsilon\}$ and $k \in \{2,\ldots,\bar{k}-1\}$ on the event $E_8 \cap \bigcap_{j=0}^{4} E_j$. A union bound implies that E_8 has probability at least

$$\begin{aligned}1 - \sum_{\bar{k}=3}^{k_\varepsilon}\left(\frac{\delta}{256k_\varepsilon} + \sum_{k=2}^{\bar{k}-1}\left(3\frac{\delta}{256k_\varepsilon^2} + \sum_{j=1}^{\tilde{j}_{\bar{k}}-1}\frac{\delta}{256k_\varepsilon^2\tilde{j}_{\bar{k}}}\right)\right) &\\ \geq 1 - \frac{\delta}{256} - \sum_{\bar{k}=3}^{k_\varepsilon}(\bar{k}-2)\frac{\delta}{64k_\varepsilon^2} \geq 1 - \frac{\delta}{256} - \frac{\delta}{128} > 1 - \frac{\delta}{64}.\end{aligned}$$

■

We can now state a sufficient size on the budget n so that, with high probability, Algorithm 1 reaches $m = \tilde{m}$, so that the returned $\hat{h}_n$ is equivalent to the $\hat{h}_\infty$ classifier from Lemma 38, which therefore satisfies the same guarantee on its error rate.

Lemma 41 *There exists a finite universal constant $\bar{c} \geq 1$ such that, on the event $\bigcap_{j=0}^{8} E_j$, for any $\bar{k} \in \{2,\ldots,k_\varepsilon\}$, for any n of size at least*

$$\begin{aligned}&\bar{c}\mathbb{1}[\bar{k}>2]2^{2\bar{k}}\left(\mathfrak{s}\mathrm{Log}\left(\frac{1}{\varepsilon}\right) + \mathrm{Log}\left(\frac{1}{\delta}\right)\right)\mathrm{Log}\left(\frac{d}{\varepsilon\delta}\right)\mathrm{Log}\left(\frac{1}{\varepsilon}\right)\\ &+ \bar{c}\sum_{k=\bar{k}}^{k_\varepsilon}\max\left\{\mathcal{P}\left(x : \gamma_x < 2^{3-k}\right), \frac{\varepsilon}{\hat{\gamma}_\varepsilon}\right\}\frac{2^k}{\varepsilon}\left(d\mathrm{Log}\left(\frac{1}{\varepsilon}\right) + \mathrm{Log}\left(\frac{1}{\delta}\right)\right)\mathrm{Log}\left(\frac{d}{\varepsilon\delta}\right)\mathrm{Log}\left(\frac{1}{\hat{\gamma}_\varepsilon}\right), \quad (36)\end{aligned}$$

running Algorithm 1 with budget n results in at most n label requests, and the returned classifier $\hat{h}_n$ satisfies $\mathrm{er}_{\mathcal{P}_{XY}}(\hat{h}_n) - \mathrm{er}_{\mathcal{P}_{XY}}(f^\star_{\mathcal{P}_{XY}}) \leq \varepsilon$. Furthermore, the event $\bigcap_{j=0}^{8} E_j$ has probability at least $1-\delta$.

Proof The value of t keeps the running total of the number of label requests made by the algorithm after each call to Subroutine 1. Furthermore, within each execution of Subroutine 1, the value $t+q$ represents the running total of the number of label requests made by the algorithm so far. Since the $n-t$ budget argument to Subroutine 1 ensures that it halts (in Step 6) if ever $t+q=n$, and since the first condition in Step 1 of Algorithm 1 ensures that Algorithm 1 halts if ever $t=n$, we are guaranteed that the algorithm never requests a number of labels larger than the budget n.

We will show that taking n of the stated size suffices for the result by showing that this size suffices to reproduce the behavior of the infinite budget execution of Algorithm 1. Due to the condition $m < \tilde{m}$ in Step 1 of Algorithm 1, the final value of t obtained when running Algorithm 1 with budget ∞ may be expressed as

$$\sum_{m=1}^{\tilde{m}} \hat{q}_{\infty,m} \mathbb{1}_{\mathrm{DIS}(V_{m-1})}\left(X_m^2\right).$$

Lemma 35 implies that, on $\bigcap_{j=0}^{8} E_j$, this is at most

$$\begin{aligned}
&\sum_{m=1}^{\tilde{m}} \left\lceil \frac{8}{\max\{\gamma_{J_m}^2, 2^{-2\tilde{k}_m}\}} \ln\left(\frac{32\tilde{m}q_{\varepsilon,\delta}}{\delta}\right) \right\rceil \mathbb{1}_{\mathrm{DIS}(V_{m-1})}\left(X_m^2\right) \\
&\leq \sum_{m=1}^{\tilde{m}} \sum_{k=2}^{\tilde{k}_m} \mathbb{1}\left[\gamma_{J_m} \leq 2^{1-k}\right] 2^{2k+4} \ln\left(\frac{32\tilde{m}q_{\varepsilon,\delta}}{\delta}\right) \mathbb{1}_{\mathrm{DIS}(V_{m-1})}\left(X_m^2\right).
\end{aligned}$$

The summation in this last expression is over all $m \in \{1,\ldots,\tilde{m}\}$ and $k \in \{2,\ldots,k_\varepsilon\}$ such that $k \leq \tilde{k}_m$, which is equivalent to those $m \in \{1,\ldots,\tilde{m}\}$ and $k \in \{2,\ldots,k_\varepsilon\}$ such that $m \leq \tilde{m}_k$. Therefore, exchanging the order of summation, this expression is equal to

$$\begin{aligned}
&\sum_{k=2}^{k_\varepsilon} \sum_{m=1}^{\tilde{m}_k} \mathbb{1}\left[\gamma_{J_m} \leq 2^{1-k}\right] 2^{2k+4} \ln\left(\frac{32\tilde{m}q_{\varepsilon,\delta}}{\delta}\right) \mathbb{1}_{\mathrm{DIS}(V_{m-1})}\left(X_m^2\right) \\
&= \sum_{k=2}^{k_\varepsilon} 2^{2k+4} \ln\left(\frac{32\tilde{m}q_{\varepsilon,\delta}}{\delta}\right) \left|\left\{m \in \{1,\ldots,\tilde{m}_k\} : \gamma_{J_m} \leq 2^{1-k}, X_m^2 \in \mathrm{DIS}(V_{m-1})\right\}\right|. \qquad (37)
\end{aligned}$$

Fix any value $\bar{k} \in \{2,\ldots,k_\varepsilon\}$. For any $k \in \left\{\bar{k},\ldots,k_\varepsilon\right\}$, Lemma 39 implies that, on $\bigcap_{j=0}^{8} E_j$,

$$\begin{aligned}
&\left|\left\{m \in \{1,\ldots,\tilde{m}_k\} : \gamma_{J_m} \leq 2^{1-k}, X_m^2 \in \mathrm{DIS}(V_{m-1})\right\}\right| \\
&\qquad \leq 17 \max\left\{\mathcal{P}\left(x : \gamma_x < 2^{3-k}\right), \frac{\varepsilon}{2\hat{\gamma}_\varepsilon}\right\} \tilde{m}_k.
\end{aligned}$$

This implies

$$\begin{aligned}
&\sum_{k=\bar{k}}^{k_\varepsilon} 2^{2k+4} \ln\left(\frac{32\tilde{m}q_{\varepsilon,\delta}}{\delta}\right) \left|\left\{m \in \{1,\ldots,\tilde{m}_k\} : \gamma_{J_m} \le 2^{1-k}, X_m^2 \in \mathrm{DIS}(V_{m-1})\right\}\right| \\
&\le \sum_{k=\bar{k}}^{k_\varepsilon} 2^{2k+9} \ln\left(\frac{32\tilde{m}q_{\varepsilon,\delta}}{\delta}\right) \max\left\{\mathcal{P}\left(x : \gamma_x < 2^{3-k}\right), \frac{\varepsilon}{2\hat{\gamma}_\varepsilon}\right\} \tilde{m}_k \\
&\le \sum_{k=\bar{k}}^{k_\varepsilon} \max\left\{\mathcal{P}\Big(x : \gamma_x < 2^{3-k}\Big), \frac{\varepsilon}{2\hat{\gamma}_\varepsilon}\right\} \frac{2^{k+17} c k_\varepsilon}{\varepsilon} \left(d\mathrm{Log}\left(\frac{2k_\varepsilon}{\varepsilon}\right) + \mathrm{Log}\left(\frac{64k_\varepsilon}{\delta}\right)\right) \mathrm{Log}\left(\frac{32\tilde{m}q_{\varepsilon,\delta}}{\delta}\right) \\
&\le \sum_{k=\bar{k}}^{k_\varepsilon} \max\left\{\mathcal{P}\Big(x : \gamma_x < 2^{3-k}\Big), \frac{\varepsilon}{2\hat{\gamma}_\varepsilon}\right\} \frac{2^{k+25} c \mathrm{Log}\left(\frac{1}{\hat{\gamma}_\varepsilon}\right)}{\varepsilon} \left(d\mathrm{Log}\left(\frac{64}{\varepsilon}\right) + \mathrm{Log}\left(\frac{1}{\delta}\right)\right) \mathrm{Log}\left(\frac{32cd}{\varepsilon\delta}\right),
\end{aligned} \tag{38}$$

where this last inequality is based on the fact that $k_\varepsilon \le \sqrt{32/\varepsilon}$, combined with some simple algebra. If $\bar{k} > 2$, for any $k \in \left\{2,\ldots,\bar{k}-1\right\}$, Lemma 40 implies that, on $\bigcap_{j=0}^{8} E_j$,

$$\begin{aligned}
&\left|\left\{m \in \{1,\ldots,\tilde{m}_k\} : \gamma_{J_m} \le 2^{1-k}, X_m^2 \in \mathrm{DIS}(V_{m-1})\right\}\right| \\
&\qquad \le 6e \max\left\{\mathcal{P}\left(x : \gamma_x < 2^{2-\bar{k}}\right), \frac{\varepsilon}{2\gamma_\varepsilon}\right\} \tilde{m}_k \\
&\qquad\quad + 91\tilde{c}\left(2^{1+\bar{k}-k} + \mathrm{Log}\left(\frac{64c}{\varepsilon}\right)\right)\left(6\mathfrak{s}\mathrm{Log}\left(\frac{128c}{\varepsilon}\right) + \mathrm{Log}\left(\frac{1}{\delta}\right)\right).
\end{aligned}$$

This implies

$$\begin{aligned}
&\sum_{k=2}^{\bar{k}-1} 2^{2k+4} \ln\left(\frac{32\tilde{m}q_{\varepsilon,\delta}}{\delta}\right) \left|\left\{m \in \{1,\ldots,\tilde{m}_k\} : \gamma_{J_m} \le 2^{1-k}, X_m^2 \in \mathrm{DIS}(V_{m-1})\right\}\right| \\
&\le \sum_{k=2}^{\bar{k}-1} 2^{2k+9} \ln\left(\frac{32\tilde{m}q_{\varepsilon,\delta}}{\delta}\right) \max\left\{\mathcal{P}\left(x : \gamma_x < 2^{2-\bar{k}}\right), \frac{\varepsilon}{2\gamma_\varepsilon}\right\} \tilde{m}_k \\
&+ \sum_{k=2}^{\bar{k}-1} 2^{2k+11} \tilde{c} \ln\left(\frac{32\tilde{m}q_{\varepsilon,\delta}}{\delta}\right)\left(2^{1+\bar{k}-k} + \mathrm{Log}\left(\frac{64c}{\varepsilon}\right)\right)\left(6\mathfrak{s}\mathrm{Log}\left(\frac{128c}{\varepsilon}\right) + \mathrm{Log}\left(\frac{1}{\delta}\right)\right).
\end{aligned}$$

Since

$$\begin{aligned}
\sum_{k=2}^{\bar{k}-1} 2^{2k} \tilde{m}_k &\le \sum_{k=2}^{\bar{k}-1} \frac{2^{k+8} c k_\varepsilon}{\varepsilon}\left(d\mathrm{Log}\left(\frac{2k_\varepsilon}{\varepsilon}\right) + \mathrm{Log}\left(\frac{64k_\varepsilon}{\delta}\right)\right) \\
&\le \frac{2^{\bar{k}+8} c k_\varepsilon}{\varepsilon}\left(d\mathrm{Log}\left(\frac{2k_\varepsilon}{\varepsilon}\right) + \mathrm{Log}\left(\frac{64k_\varepsilon}{\delta}\right)\right) \\
&\le \frac{2^{\bar{k}+12} c \mathrm{Log}(1/\hat{\gamma}_\varepsilon)}{\varepsilon}\left(d\mathrm{Log}\left(\frac{64}{\varepsilon}\right) + \mathrm{Log}\left(\frac{1}{\delta}\right)\right)
\end{aligned}$$

and

$$\sum_{k=2}^{\bar{k}-1} 2^{2k}\left(2^{1+\bar{k}-k}+\operatorname{Log}\left(\frac{64c}{\varepsilon}\right)\right) \le 2^{2\bar{k}}\left(2+\operatorname{Log}\left(\frac{64c}{\varepsilon}\right)\right) \le 2^{2\bar{k}+1}\operatorname{Log}\left(\frac{64c}{\varepsilon}\right),$$

we have that

$$\begin{aligned}
&\sum_{k=2}^{\bar{k}-1} 2^{2k+4}\ln\left(\frac{32\tilde{m}q_{\varepsilon,\delta}}{\delta}\right)\left|\left\{m\in\{1,\dots,\tilde{m}_k\}:\gamma_{J_m}\le 2^{1-k}, X_m^2\in \mathrm{DIS}(V_{m-1})\right\}\right| \\
&\le 2^9\ln\left(\frac{32\tilde{m}q_{\varepsilon,\delta}}{\delta}\right)\max\left\{\mathcal{P}\left(x:\gamma_x<2^{2-\bar{k}}\right),\frac{\varepsilon}{2\gamma_\varepsilon}\right\}\sum_{k=2}^{\bar{k}-1}2^{2k}\tilde{m}_k \\
&\quad + 2^{11}\tilde{c}\ln\left(\frac{32\tilde{m}q_{\varepsilon,\delta}}{\delta}\right)\left(6\mathfrak{s}\operatorname{Log}\left(\frac{128c}{\varepsilon}\right)+\operatorname{Log}\left(\frac{1}{\delta}\right)\right)\sum_{k=2}^{\bar{k}-1}2^{2k}\left(2^{1+\bar{k}-k}+\operatorname{Log}\left(\frac{64c}{\varepsilon}\right)\right) \\
&\le 2^9\ln\left(\frac{32\tilde{m}q_{\varepsilon,\delta}}{\delta}\right)\max\left\{\mathcal{P}\left(x:\gamma_x<2^{2-\bar{k}}\right),\frac{\varepsilon}{2\gamma_\varepsilon}\right\}\frac{2^{\bar{k}+12}c\operatorname{Log}\left(\frac{1}{\tilde{\gamma}_\varepsilon}\right)}{\varepsilon}\left(d\operatorname{Log}\left(\frac{64}{\varepsilon}\right)+\operatorname{Log}\left(\frac{1}{\delta}\right)\right) \\
&\quad + 2^{11}\tilde{c}\ln\left(\frac{32\tilde{m}q_{\varepsilon,\delta}}{\delta}\right)\left(6\mathfrak{s}\operatorname{Log}\left(\frac{128c}{\varepsilon}\right)+\operatorname{Log}\left(\frac{1}{\delta}\right)\right)2^{2\bar{k}+1}\operatorname{Log}\left(\frac{64c}{\varepsilon}\right) \\
&\le \max\left\{\mathcal{P}\left(x:\gamma_x<2^{2-\bar{k}}\right),\frac{\varepsilon}{2\gamma_\varepsilon}\right\}\frac{2^{\bar{k}+25}c\operatorname{Log}\left(\frac{1}{\tilde{\gamma}_\varepsilon}\right)}{\varepsilon}\left(d\operatorname{Log}\left(\frac{64}{\varepsilon}\right)+\operatorname{Log}\left(\frac{1}{\delta}\right)\right)\operatorname{Log}\left(\frac{32cd}{\varepsilon\delta}\right) \\
&\quad + 2^{2\bar{k}+16}\tilde{c}\left(6\mathfrak{s}\operatorname{Log}\left(\frac{128c}{\varepsilon}\right)+\operatorname{Log}\left(\frac{1}{\delta}\right)\right)\operatorname{Log}\left(\frac{64c}{\varepsilon}\right)\operatorname{Log}\left(\frac{32cd}{\varepsilon\delta}\right).
\end{aligned}$$

Plugging this and (38) into (37) reveals that, on $\bigcap_{j=0}^{8} E_j$, if $\bar{k}>2$,

$$\begin{aligned}
&\sum_{m=1}^{\tilde{m}}\hat{q}_{\infty,m}\mathbb{1}_{\mathrm{DIS}(V_{m-1})}\left(X_m^2\right) \\
&\le \max\left\{\mathcal{P}\left(x:\gamma_x<2^{2-\bar{k}}\right),\frac{\varepsilon}{2\gamma_\varepsilon}\right\}\frac{2^{\bar{k}+25}c\operatorname{Log}\left(\frac{1}{\tilde{\gamma}_\varepsilon}\right)}{\varepsilon}\left(d\operatorname{Log}\left(\frac{64}{\varepsilon}\right)+\operatorname{Log}\left(\frac{1}{\delta}\right)\right)\operatorname{Log}\left(\frac{32cd}{\varepsilon\delta}\right) \\
&+ 2^{2\bar{k}+16}\tilde{c}\left(6\mathfrak{s}\operatorname{Log}\left(\frac{128c}{\varepsilon}\right)+\operatorname{Log}\left(\frac{1}{\delta}\right)\right)\operatorname{Log}\left(\frac{64c}{\varepsilon}\right)\operatorname{Log}\left(\frac{32cd}{\varepsilon\delta}\right) \\
&+ \sum_{k=\bar{k}}^{k_\varepsilon}\max\left\{\mathcal{P}\left(x:\gamma_x<2^{3-k}\right),\frac{\varepsilon}{2\hat{\gamma}_\varepsilon}\right\}\frac{2^{k+25}c\operatorname{Log}\left(\frac{1}{\tilde{\gamma}_\varepsilon}\right)}{\varepsilon}\left(d\operatorname{Log}\left(\frac{64}{\varepsilon}\right)+\operatorname{Log}\left(\frac{1}{\delta}\right)\right)\operatorname{Log}\left(\frac{32cd}{\varepsilon\delta}\right). \\
&\le \bar{c}2^{2\bar{k}}\left(\mathfrak{s}\operatorname{Log}\left(\frac{1}{\varepsilon}\right)+\operatorname{Log}\left(\frac{1}{\delta}\right)\right)\operatorname{Log}\left(\frac{d}{\varepsilon\delta}\right)\operatorname{Log}\left(\frac{1}{\varepsilon}\right) \\
&+ \bar{c}\sum_{k=\bar{k}}^{k_\varepsilon}\max\left\{\mathcal{P}\left(x:\gamma_x<2^{3-k}\right),\frac{\varepsilon}{\hat{\gamma}_\varepsilon}\right\}\frac{2^k}{\varepsilon}\left(d\operatorname{Log}\left(\frac{1}{\varepsilon}\right)+\operatorname{Log}\left(\frac{1}{\delta}\right)\right)\operatorname{Log}\left(\frac{d}{\varepsilon\delta}\right)\operatorname{Log}\left(\frac{1}{\hat{\gamma}_\varepsilon}\right),
\end{aligned}$$

for an appropriate finite universal constant $\bar{c} \geq 1$. Furthermore, if $\bar{k} = 2$, (38) and (37) already imply that, on $\bigcap_{j=0}^{8} E_j$,

$$\sum_{m=1}^{\tilde{m}} \hat{q}_{\infty,m} \mathbb{1}_{\mathrm{DIS}(V_{m-1})}\left(X_m^2\right)$$
$$\leq \bar{c} \sum_{k=\bar{k}}^{k_\varepsilon} \max\left\{\mathcal{P}\left(x : \gamma_x < 2^{3-k}\right), \frac{\varepsilon}{\hat{\gamma}_\varepsilon}\right\} \frac{2^k}{\varepsilon}\left(d\mathrm{Log}\left(\frac{1}{\varepsilon}\right) + \mathrm{Log}\left(\frac{1}{\delta}\right)\right) \mathrm{Log}\left(\frac{d}{\varepsilon\delta}\right) \mathrm{Log}\left(\frac{1}{\hat{\gamma}_\varepsilon}\right),$$

again for $\bar{c} \geq 1$ chosen appropriately large.

Therefore, for a choice of $\bar{c}$ as above, on $\bigcap_{j=0}^{8} E_j$, for any $\bar{k} \in \{2, \ldots, k_\varepsilon\}$, the final value of t obtained when running Algorithm 1 with budget ∞ is at most (36). Since running Algorithm 1 with a finite budget n only returns a different $\hat{h}_n$ from the $\hat{h}_\infty$ returned by the infinite-budget execution if t would exceed n in the infinite-budget execution, this implies that taking any n of size at least (36) suffices to produce identical output to the infinite-budget execution, on the event $\bigcap_{j=0}^{8} E_j$: that is, $\hat{h}_n = \hat{h}_\infty$. Therefore, since Lemma 38 implies that, on $\bigcap_{j=0}^{8} E_j$, $\mathrm{er}_{\mathcal{P}_{XY}}(\hat{h}_\infty) - \mathrm{er}_{\mathcal{P}_{XY}}(f^\star_{\mathcal{P}_{XY}}) \leq \varepsilon$, we conclude that for n of size at least (36), on $\bigcap_{j=0}^{8} E_j$, $\mathrm{er}_{\mathcal{P}_{XY}}(\hat{h}_n) - \mathrm{er}_{\mathcal{P}_{XY}}(f^\star_{\mathcal{P}_{XY}}) \leq \varepsilon$.

Finally, by a union bound, the event $\bigcap_{j=0}^{8} E_j$ has probability at least

$$1 - 0 - \frac{\delta}{2} - \frac{\delta}{512} - \frac{\delta}{4} - \frac{\delta}{32} - 4\frac{\delta}{64} > 1 - \delta.$$

■

We can obtain the upper bounds for Theorems 4, 5, and 7 from Section 5 by straightforward applications of Lemma 41. Note that, due to the choice of $\hat{\gamma}_\varepsilon$ in each of these proofs, Algorithm 1 is not adaptive to the noise parameters. It is conceivable that this dependence can be removed by a model selection procedure (see Balcan and Hanneke, 2012; Hanneke, 2011, for discussions related to this). However, we do not discuss this further here, leaving this important issue for future work. The upper bounds for Theorems 6 and 8 are based on known results for other algorithms in the literature, though the lower bound for Theorem 6 is new here. The remainder of this section provides the details of these proofs.

Proof of Theorem 4 Fix any $\beta \in [0, 1/2)$, $\varepsilon, \delta \in (0, 1)$, and $\mathcal{P}_{XY} \in \mathrm{BN}(\beta)$. Any $\gamma < 1/2 - \beta$ has $\mathcal{P}(x : \gamma_x \leq \gamma) = 0$, and since we always have $\gamma_\varepsilon \geq \varepsilon/2$, we must have $\gamma_\varepsilon \geq \max\{1/2 - \beta, \varepsilon/2\}$. We may therefore take $\hat{\gamma}_\varepsilon = \max\{1/2 - \beta, \varepsilon/2\}$. Therefore, taking $\bar{k} = k_\varepsilon$ in Lemma 41, the first term in (36) is at most

$$\frac{2^{10}\bar{c}}{(1-2\beta)^2}\left(\mathfrak{s}\mathrm{Log}\left(\frac{1}{\varepsilon}\right) + \mathrm{Log}\left(\frac{1}{\delta}\right)\right) \mathrm{Log}\left(\frac{d}{\varepsilon\delta}\right) \mathrm{Log}\left(\frac{1}{\varepsilon}\right),$$

while the second term in (36) is at most

$$\bar{c} \max\left\{\mathcal{P}\left(x : \gamma_x < \hat{\gamma}_\varepsilon\right), \frac{\varepsilon}{\hat{\gamma}_\varepsilon}\right\} \frac{16}{\hat{\gamma}_\varepsilon \varepsilon}\left(d\mathrm{Log}\left(\frac{1}{\varepsilon}\right) + \mathrm{Log}\left(\frac{1}{\delta}\right)\right) \mathrm{Log}\left(\frac{d}{\varepsilon\delta}\right) \mathrm{Log}\left(\frac{1}{\hat{\gamma}_\varepsilon}\right).$$

Since $\mathcal{P}(x : \gamma_x < 1/2 - \beta) = 0 < \frac{\varepsilon}{1/2-\beta}$ and $\mathcal{P}(x : \gamma_x < \varepsilon/2) \leq 1 < 2 = \frac{\varepsilon}{\varepsilon/2}$, we have that $\mathcal{P}(x : \gamma_x < \hat{\gamma}_\varepsilon) < \frac{\varepsilon}{\hat{\gamma}_\varepsilon}$, so that the above is at most

$$\frac{64\bar{c}}{(1-2\beta)^2}\left(d\mathrm{Log}\left(\frac{1}{\varepsilon}\right)+\mathrm{Log}\left(\frac{1}{\delta}\right)\right)\mathrm{Log}\left(\frac{d}{\varepsilon\delta}\right)\mathrm{Log}\left(\frac{2}{(1-2\beta)\vee\varepsilon}\right).$$

Therefore, recalling that $\mathfrak{s} \geq d$, since Lemma 41 implies that, with any budget n at least the size of the sum of these two terms, Algorithm 1 produces a classifier $\hat{h}_n$ with $\mathrm{er}_{\mathcal{P}_{XY}}(\hat{h}_n) - \mathrm{er}_{\mathcal{P}_{XY}}(f^\star_{\mathcal{P}_{XY}}) \leq \varepsilon$ with probability at least $1-\delta$, and requests a number of labels at most n, we have that

$$\begin{aligned}\Lambda_{\mathrm{BN}(\beta)}(\varepsilon,\delta) &\leq \frac{2^{10}\bar{c}}{(1-2\beta)^2}\left(\mathfrak{s}\mathrm{Log}\left(\frac{1}{\varepsilon}\right)+\mathrm{Log}\left(\frac{1}{\delta}\right)\right)\mathrm{Log}\left(\frac{d}{\varepsilon\delta}\right)\mathrm{Log}\left(\frac{1}{\varepsilon}\right)\\ &\quad+\frac{64\bar{c}}{(1-2\beta)^2}\left(d\mathrm{Log}\left(\frac{1}{\varepsilon}\right)+\mathrm{Log}\left(\frac{1}{\delta}\right)\right)\mathrm{Log}\left(\frac{d}{\varepsilon\delta}\right)\mathrm{Log}\left(\frac{2}{(1-2\beta)\vee\varepsilon}\right)\\ &\lesssim \frac{1}{(1-2\beta)^2}\left(\mathfrak{s}\mathrm{Log}\left(\frac{1}{\varepsilon}\right)+\mathrm{Log}\left(\frac{1}{\delta}\right)\right)\mathrm{Log}\left(\frac{d}{\varepsilon\delta}\right)\mathrm{Log}\left(\frac{1}{\varepsilon}\right).\end{aligned}$$

On the other hand, Giné and Koltchinskii (2006) have shown that for the passive learning method of *empirical risk minimization*, producing a classifier $\check{h}_n$ satisfying $\check{h}_n = \mathrm{argmin}_{h\in\mathbb{C}}\sum_{m=1}^n \mathbb{1}[h(X_m) \neq Y_m]$, if n is of size at least

$$\frac{\check{c}}{(1-2\beta)\varepsilon}\left(d\mathrm{Log}\left(\theta_{\mathcal{P}_{XY}}\left(\frac{\varepsilon}{1-2\beta}\right)\right)+\mathrm{Log}\left(\frac{1}{\delta}\right)\right),$$

for an appropriate finite universal constant $\check{c}$, then with probability at least $1-\delta$, we have $\mathrm{er}_{\mathcal{P}_{XY}}(\check{h}_n) - \mathrm{er}_{\mathcal{P}_{XY}}(f^\star_{\mathcal{P}_{XY}}) \leq \varepsilon$. Therefore, since Theorem 10 implies $\theta_{\mathcal{P}_{XY}}(\varepsilon/(1-2\beta)) \leq \theta_{\mathcal{P}_{XY}}((\varepsilon/(1-2\beta)) \wedge 1) \leq \min\left\{\mathfrak{s}, \frac{1-2\beta}{\varepsilon}\vee 1\right\}$, it follows that

$$\Lambda_{\mathrm{BN}(\beta)}(\varepsilon,\delta) \lesssim \frac{1}{(1-2\beta)\varepsilon}\left(d\mathrm{Log}\left(\min\left\{\mathfrak{s}, \frac{1-2\beta}{\varepsilon}\right\}\right)+\mathrm{Log}\left(\frac{1}{\delta}\right)\right).$$

Together, these two bounds on $\Lambda_{\mathrm{BN}(\beta)}(\varepsilon,\delta)$ imply the following upper bound, simply by choosing whichever of these two methods has the smaller corresponding bound for the given values of ε, δ, β, d, and $\mathfrak{s}$.

$$\Lambda_{\mathrm{BN}(\beta)}(\varepsilon,\delta) \lesssim \min\left\{\begin{array}{l}\frac{1}{(1-2\beta)^2}\left(\mathfrak{s}\mathrm{Log}\left(\frac{1}{\varepsilon}\right)+\mathrm{Log}\left(\frac{1}{\delta}\right)\right)\mathrm{Log}\left(\frac{d}{\varepsilon\delta}\right)\mathrm{Log}\left(\frac{1}{\varepsilon}\right)\\ \frac{1}{(1-2\beta)\varepsilon}\left(d\mathrm{Log}\left(\min\left\{\mathfrak{s}, \frac{1-2\beta}{\varepsilon}\right\}\right)+\mathrm{Log}\left(\frac{1}{\delta}\right)\right)\end{array}\right..$$

The statement of the upper bound in Theorem 4 represents a relaxation of this, in that it is slightly larger (in the logarithmic factors), the intention being that it is a simpler expression to state. To arrive at this relaxation, we note that $\mathfrak{s}\mathrm{Log}\left(\frac{1}{\varepsilon}\right)+\mathrm{Log}\left(\frac{1}{\delta}\right) \leq \mathfrak{s}\mathrm{Log}\left(\frac{1}{\varepsilon\delta}\right)$, and $d\mathrm{Log}\left(\min\left\{\mathfrak{s}, \frac{1-2\beta}{\varepsilon}\right\}\right)+\mathrm{Log}\left(\frac{1}{\delta}\right) \leq d\mathrm{Log}\left(\frac{1}{\varepsilon\delta}\right)\mathrm{Log}\left(\frac{d}{\varepsilon\delta}\right)\mathrm{Log}\left(\frac{1}{\varepsilon}\right)$, so that the above is at most

$$\frac{1}{(1-2\beta)^2}\min\left\{\mathfrak{s}, \frac{(1-2\beta)d}{\varepsilon}\right\}\mathrm{Log}\left(\frac{d}{\varepsilon\delta}\right)\mathrm{Log}\left(\frac{1}{\varepsilon\delta}\right)\mathrm{Log}\left(\frac{1}{\varepsilon}\right).$$

Next, we turn to establishing the lower bound. Fix $\varepsilon \in (0, (1-2\beta)/24)$ and $\delta \in (0, 1/24]$. First note that taking $\zeta = \frac{2\varepsilon}{1-2\beta}$ and $k = \min\{\mathfrak{s}-1, \lfloor 1/\zeta \rfloor\}$ in Lemma 26, we have $\mathrm{RR}(k,\zeta,\beta) \subseteq \mathrm{BN}(\beta)$, so that Lemma 26 implies

$$\Lambda_{\mathrm{BN}(\beta)}(\varepsilon,\delta) \geq \Lambda_{\mathrm{RR}(k,\zeta,\beta)}(\varepsilon,\delta) = \Lambda_{\mathrm{RR}(k,\zeta,\beta)}((\zeta/2)(1-2\beta),\delta) \geq \frac{\beta(k-1)\ln\left(\frac{1}{4\delta}\right)}{3(1-2\beta)^2}$$
$$\geq \min\left\{\mathfrak{s}-2, \frac{1-2\zeta}{\zeta}\right\}\frac{\beta\ln\left(\frac{1}{4\delta}\right)}{3(1-2\beta)^2} = \frac{\beta}{(1-2\beta)^2}\min\left\{\mathfrak{s}-2, \frac{1-2\beta-4\varepsilon}{2\varepsilon}\right\}\ln\left(\frac{1}{4\delta}\right).$$
$$\geq \frac{\beta}{8(1-2\beta)^2}\min\left\{\mathfrak{s}-2, \frac{1-2\beta}{\varepsilon}\right\}\mathrm{Log}\left(\frac{1}{\delta}\right). \quad (39)$$

Additionally, based on techniques of Kääriäinen (2006); Beygelzimer, Dasgupta, and Langford (2009); Hanneke (2011), the recent article of Hanneke (2014) contains the following lower bound (in the proof of Theorem 4.3 there), for $\varepsilon \in (0,(1-2\beta)/24)$ and $\delta \in (0,1/24]$.

$$\Lambda_{\mathrm{BN}(\beta)}(\varepsilon,\delta) \geq \max\left\{2\left\lfloor\frac{1-(1-2\beta)^2}{2(1-2\beta)^2}\ln\left(\frac{1}{8\delta(1-2\delta)}\right)\right\rfloor, \frac{d-1}{6}\left\lfloor\frac{1-(1-2\beta)^2}{2(1-2\beta)^2}\ln\left(\frac{9}{8}\right)\right\rfloor\right\}$$
$$\geq \max\left\{2\left\lfloor\frac{\beta}{(1-2\beta)^2}\mathrm{Log}\left(\frac{1}{8\delta}\right)\right\rfloor, \frac{d-1}{6}\left\lfloor\frac{\beta}{10(1-2\beta)^2}\right\rfloor\right\}.$$

If $\frac{\beta}{(1-2\beta)^2}\mathrm{Log}\left(\frac{1}{8\delta}\right) \geq 1$, then $2\left\lfloor\frac{\beta}{(1-2\beta)^2}\mathrm{Log}\left(\frac{1}{8\delta}\right)\right\rfloor \geq \frac{\beta}{(1-2\beta)^2}\mathrm{Log}\left(\frac{1}{8\delta}\right) \geq \frac{\beta}{3(1-2\beta)^2}\mathrm{Log}\left(\frac{1}{\delta}\right)$, so that $\Lambda_{\mathrm{BN}(\beta)}(\varepsilon,\delta) \gtrsim \frac{\beta}{(1-2\beta)^2}\mathrm{Log}\left(\frac{1}{\delta}\right)$. Otherwise, if $\frac{\beta}{(1-2\beta)^2}\mathrm{Log}\left(\frac{1}{8\delta}\right) < 1$, then since $\mathrm{RE} \subseteq \mathrm{BN}(\beta)$, and $|\mathbb{C}| \geq 2$ implies $d \geq 1 > \frac{\beta}{(1-2\beta)^2}\mathrm{Log}\left(\frac{1}{8\delta}\right)$, Theorem 3 (proven above) implies we still have $\Lambda_{\mathrm{BN}(\beta)}(\varepsilon,\delta) \geq \Lambda_{\mathrm{RE}}(\varepsilon,\delta) \gtrsim \frac{\beta}{(1-2\beta)^2}\mathrm{Log}\left(\frac{1}{\delta}\right)$ in this case. When $d=1$, these observations further imply $\Lambda_{\mathrm{BN}(\beta)} \gtrsim \frac{d\beta}{(1-2\beta)^2}$. On the other hand, if $d>1$, and if $\frac{\beta}{10(1-2\beta)^2} \geq 1$, then $\frac{d-1}{6}\left\lfloor\frac{\beta}{10(1-2\beta)^2}\right\rfloor \geq \frac{d}{240}\frac{\beta}{(1-2\beta)^2}$, so that $\Lambda_{\mathrm{BN}(\beta)}(\varepsilon,\delta) \gtrsim \frac{d\beta}{(1-2\beta)^2}$. Otherwise, if $\frac{\beta}{10(1-2\beta)^2} < 1$, then since $\mathrm{RE} \subseteq \mathrm{BN}(\beta)$, Theorem 3 implies we still have $\Lambda_{\mathrm{BN}(\beta)}(\varepsilon,\delta) \geq \Lambda_{\mathrm{RE}}(\varepsilon,\delta) \gtrsim d \gtrsim \frac{d\beta}{(1-2\beta)^2}$ in this case as well. If $\beta > 1/4$, then $\frac{d\beta}{(1-2\beta)^2} \geq \frac{d}{4(1-2\beta)^2} \gtrsim \frac{d}{(1-2\beta)^2}$, so that $\Lambda_{\mathrm{BN}(\beta)}(\varepsilon,\delta) \gtrsim \frac{d}{(1-2\beta)^2}$. Otherwise, if $\beta \leq 1/4$, then $\frac{1}{(1-2\beta)^2} \leq 4$, so that Theorem 3 implies $\Lambda_{\mathrm{BN}(\beta)}(\varepsilon,\delta) \geq \Lambda_{\mathrm{RE}}(\varepsilon,\delta) \gtrsim d \gtrsim \frac{d}{(1-2\beta)^2}$. Altogether, we have that

$$\Lambda_{\mathrm{BN}(\beta)}(\varepsilon,\delta) \gtrsim \frac{1}{(1-2\beta)^2}\max\left\{\beta\mathrm{Log}\left(\frac{1}{\delta}\right), d\right\}. \quad (40)$$

When $\mathfrak{s} \leq 2$, $\min\left\{\mathfrak{s}, \frac{1-2\beta}{\varepsilon}\right\} \leq 2$, so that (40) trivially implies

$$\Lambda_{\mathrm{BN}(\beta)}(\varepsilon,\delta) \gtrsim \frac{1}{(1-2\beta)^2}\max\left\{\min\left\{\mathfrak{s}, \frac{1-2\beta}{\varepsilon}\right\}\beta\mathrm{Log}\left(\frac{1}{\delta}\right), d\right\}. \quad (41)$$

Otherwise, when $\mathfrak{s} \geq 3$, we have $\mathfrak{s}-2 \geq \mathfrak{s}/3$, so that $\min\left\{\mathfrak{s}-2, \frac{1-2\beta}{\varepsilon}\right\} \geq \frac{1}{3}\min\left\{\mathfrak{s}, \frac{1-2\beta}{\varepsilon}\right\}$. Combined with (39) and (40), this implies (41) holds in this case as well. ■

Proof of Theorem 5 We begin with the upper bounds. Fix any $a \in [1,\infty)$, $\alpha \in (0,1)$, $\varepsilon, \delta \in (0,1)$, and $\mathcal{P}_{XY} \in \mathrm{TN}(a,\alpha)$. For any $\gamma \le \left(\frac{\varepsilon}{2a'}\right)^{1-\alpha}$, by definition of $\mathrm{TN}(a,\alpha)$, we have $\gamma \mathcal{P}\left(x : \gamma_x \le \gamma\right) \le a'\gamma^{1/(1-\alpha)} \le \varepsilon/2$. Therefore, since we always have $\gamma_\varepsilon \ge \varepsilon/2$, we have $\gamma_\varepsilon \ge \max\left\{\left(\frac{\varepsilon}{2a'}\right)^{1-\alpha}, \frac{\varepsilon}{2}\right\}$, so that we can take $\hat{\gamma}_\varepsilon = \max\left\{\left(\frac{\varepsilon}{2a'}\right)^{1-\alpha}, \frac{\varepsilon}{2}\right\}$.

Therefore, taking $\bar{k} = 2$ in Lemma 41 implies that, with any budget n of size at least

$$\bar{c}\sum_{k=2}^{k_\varepsilon} \max\left\{\min\left\{a' 2^{(3-k)\alpha/(1-\alpha)}, 1\right\}, \frac{\varepsilon}{\hat{\gamma}_\varepsilon}\right\} \frac{2^k}{\varepsilon}\left(d\mathrm{Log}\left(\frac{1}{\varepsilon}\right) + \mathrm{Log}\left(\frac{1}{\delta}\right)\right)\mathrm{Log}\left(\frac{d}{\varepsilon\delta}\right)\mathrm{Log}\left(\frac{1}{\hat{\gamma}_\varepsilon}\right), \tag{42}$$

Algorithm 1 produces a classifier $\hat{h}_n$ with $\mathrm{er}_{\mathcal{P}_{XY}}(\hat{h}_n) - \mathrm{er}_{\mathcal{P}_{XY}}(f^\star_{\mathcal{P}_{XY}}) \le \varepsilon$ with probability at least $1-\delta$, and requests a number of labels at most n. This implies $\Lambda_{\mathrm{TN}(a,\alpha)}(\varepsilon,\delta)$ is at most (42).

First note that

$$\begin{aligned}\sum_{k=2}^{k_\varepsilon} \frac{\varepsilon}{\hat{\gamma}_\varepsilon}\frac{2^k}{\varepsilon} &\le \frac{2^{1+k_\varepsilon}}{\hat{\gamma}_\varepsilon} = \frac{2^{\lceil \log_2(16/\hat{\gamma}_\varepsilon)\rceil}}{\hat{\gamma}_\varepsilon} \le \frac{32}{\hat{\gamma}_\varepsilon^2} \le 32\min\left\{\left(2a'\right)^{2-2\alpha}\varepsilon^{2\alpha-2}, 4\varepsilon^{-2}\right\} \\ &= 32\min\left\{(2-2\alpha)^{2-2\alpha}(2\alpha)^{2\alpha}a^2\varepsilon^{2\alpha-2}, 4\varepsilon^{-2}\right\} \le 128\min\left\{a^2\varepsilon^{2\alpha-2}, \varepsilon^{-2}\right\}. \end{aligned} \tag{43}$$

Furthermore, since $\varepsilon^{-2} < a^2\varepsilon^{2\alpha-2}$ only if $\varepsilon > a^{-1/\alpha}$, this is at most $128\min\left\{a^2\varepsilon^{2\alpha-2}, a^{1/\alpha}\varepsilon^{-1}\right\}$. Also, for $\alpha \ge 1/2$, letting $k_{(a,\alpha)} = \left\lceil \log_2\left(8\left(a'\right)^{(1-\alpha)/\alpha}\right)\right\rceil$, we have $k_{(a,\alpha)} \ge 2$. Additionally, for $\alpha \ge 1/2$, $2^{k\frac{1-2\alpha}{1-\alpha}}$ is nonincreasing in k. In particular, if $k_{(a,\alpha)} = 2$, then

$$\sum_{k=2}^{k_\varepsilon} \min\left\{a' 2^{(3-k)\alpha/(1-\alpha)}, 1\right\}\frac{2^k}{\varepsilon} \le \sum_{k=k_{(a,\alpha)}}^{k_\varepsilon} \frac{8a'}{\varepsilon} 2^{(k-3)\frac{1-2\alpha}{1-\alpha}} \le \frac{8k_\varepsilon}{\varepsilon}(a')^{\frac{1-\alpha}{\alpha}}.$$

Otherwise, if $k_{(a,\alpha)} \ge 3$, then

$$\begin{aligned}\sum_{k=2}^{k_\varepsilon} \min\left\{a' 2^{(3-k)\alpha/(1-\alpha)}, 1\right\}\frac{2^k}{\varepsilon} &\le \sum_{k=2}^{k_{(a,\alpha)}-1}\frac{2^k}{\varepsilon} + \sum_{k=k_{(a,\alpha)}}^{k_\varepsilon}\frac{8a'}{\varepsilon}2^{(k-3)\frac{1-2\alpha}{1-\alpha}}. \\ &\le \frac{16}{\varepsilon}(a')^{\frac{1-\alpha}{\alpha}} + \frac{8(k_\varepsilon-2)}{\varepsilon}(a')^{\frac{1-\alpha}{\alpha}} = \frac{8k_\varepsilon}{\varepsilon}(a')^{\frac{1-\alpha}{\alpha}}.\end{aligned}$$

Furthermore, since $(1-\alpha)^{\frac{1-\alpha}{\alpha}} \le 1$, we have

$$\frac{8k_\varepsilon}{\varepsilon}(a')^{\frac{1-\alpha}{\alpha}} = \frac{8k_\varepsilon}{\varepsilon}(1-\alpha)^{\frac{1-\alpha}{\alpha}}(2\alpha)a^{1/\alpha} \le \frac{16k_\varepsilon}{\varepsilon}a^{1/\alpha}.$$

Therefore, in either case, when $\alpha \ge 1/2$, (42) is at most

$$\begin{aligned}&\bar{c}\left(16k_\varepsilon a^{1/\alpha}\varepsilon^{-1} + 128a^{1/\alpha}\varepsilon^{-1}\right)\left(d\mathrm{Log}\left(\frac{1}{\varepsilon}\right) + \mathrm{Log}\left(\frac{1}{\delta}\right)\right)\mathrm{Log}\left(\frac{d}{\varepsilon\delta}\right)\mathrm{Log}\left(\frac{1}{\hat{\gamma}_\varepsilon}\right) \\ &\le 767\bar{c}\frac{a^{1/\alpha}}{\varepsilon}\left(d\mathrm{Log}\left(\frac{1}{\varepsilon}\right) + \mathrm{Log}\left(\frac{1}{\delta}\right)\right)\mathrm{Log}\left(\frac{d}{\varepsilon\delta}\right)\mathrm{Log}^2\left(\frac{1}{\varepsilon}\right),\end{aligned}$$

which is therefore an upper bound on $\Lambda_{\mathrm{TN}(a,\alpha)}(\varepsilon,\delta)$ in this case.

Otherwise, if $\alpha \le 1/2$, then $2^{k\frac{1-2\alpha}{1-\alpha}}$ is nondecreasing in k, so that

$$\sum_{k=2}^{k_\varepsilon} \min\left\{a'2^{(3-k)\alpha/(1-\alpha)}, 1\right\} \frac{2^k}{\varepsilon} \le \sum_{k=2}^{k_\varepsilon} 8a'2^{(k-3)\frac{1-2\alpha}{1-\alpha}} \frac{1}{\varepsilon} \le (k_\varepsilon - 1)8a'2^{(k_\varepsilon-3)\frac{1-2\alpha}{1-\alpha}} \frac{1}{\varepsilon}$$
$$\le (k_\varepsilon - 1)8a' \left(\frac{2}{\hat{\gamma}_\varepsilon}\right)^{\frac{1-2\alpha}{1-\alpha}} \frac{1}{\varepsilon} \le (k_\varepsilon - 1)8a'2^{\frac{1-2\alpha}{1-\alpha}} \left(2a'\right)^{1-2\alpha} \left(\frac{1}{\varepsilon}\right)^{2-2\alpha} \le (k_\varepsilon - 1)32 \left(\frac{a'}{\varepsilon}\right)^{2-2\alpha}$$
$$= (k_\varepsilon - 1)32(1-\alpha)^{2-2\alpha}(2\alpha)^{2\alpha} a^2 \varepsilon^{2\alpha-2} \le (k_\varepsilon - 1)32a^2\varepsilon^{2\alpha-2}.$$

Therefore, (42) is at most

$$\bar{c}\left((k_\varepsilon - 1)32a^2\varepsilon^{2\alpha-2} + 128a^2\varepsilon^{2\alpha-2}\right)\left(d\mathrm{Log}\left(\frac{1}{\varepsilon}\right) + \mathrm{Log}\left(\frac{1}{\delta}\right)\right)\mathrm{Log}\left(\frac{d}{\varepsilon\delta}\right)\mathrm{Log}\left(\frac{1}{\hat{\gamma}_\varepsilon}\right)$$
$$\le 832\bar{c}a^2\varepsilon^{2\alpha-2}\left(d\mathrm{Log}\left(\frac{1}{\varepsilon}\right) + \mathrm{Log}\left(\frac{1}{\delta}\right)\right)\mathrm{Log}\left(\frac{d}{\varepsilon\delta}\right)\mathrm{Log}^2\left(\frac{1}{\varepsilon}\right).$$

In particular, this implies $\Lambda_{\mathrm{TN}(a,\alpha)}(\varepsilon,\delta)$ is at most this large when $\alpha \le 1/2$. Furthermore, this completes the proof of the upper bound for the cases where either $\alpha \le 1/2$, or $\alpha \ge 1/2$ and $\frac{\mathfrak{s}}{d} \ge \frac{1}{a^{1/\alpha}\varepsilon}$.

Next, consider the remaining case that $\alpha \ge 1/2$ and $\frac{\mathfrak{s}}{d} < \frac{1}{a^{1/\alpha}\varepsilon}$. In particular, this requires that $\mathfrak{s} < \infty$, and since $\mathfrak{s} \ge d$, that $\varepsilon < a^{-1/\alpha}$. In this case, let us take

$$\bar{k} = 3 + \left\lceil (1-\alpha)\log_2\left(\frac{k_\varepsilon a'}{8\varepsilon}\frac{d\mathrm{Log}\left(\frac{1}{\varepsilon}\right) + \mathrm{Log}\left(\frac{1}{\delta}\right)}{\mathfrak{s}\mathrm{Log}\left(\frac{1}{\varepsilon}\right) + \mathrm{Log}\left(\frac{1}{\delta}\right)}\right)\right\rceil.$$

Since $\mathfrak{s} \ge d$, we have $\frac{\mathfrak{s}\mathrm{Log}\left(\frac{1}{\varepsilon}\right)+\mathrm{Log}\left(\frac{1}{\delta}\right)}{d\mathrm{Log}\left(\frac{1}{\varepsilon}\right)+\mathrm{Log}\left(\frac{1}{\delta}\right)} \le \frac{\mathfrak{s}\mathrm{Log}\left(\frac{1}{\varepsilon}\right)}{d\mathrm{Log}\left(\frac{1}{\varepsilon}\right)} = \frac{\mathfrak{s}}{d}$, so that, since $\frac{\mathfrak{s}}{d} < \frac{1}{a^{1/\alpha}\varepsilon}$, we have $\frac{\mathfrak{s}\mathrm{Log}\left(\frac{1}{\varepsilon}\right)+\mathrm{Log}\left(\frac{1}{\delta}\right)}{d\mathrm{Log}\left(\frac{1}{\varepsilon}\right)+\mathrm{Log}\left(\frac{1}{\delta}\right)} < \frac{1}{a^{1/\alpha}\varepsilon}$. A bit of algebra reveals that, in this case, $\bar{k} \ge 2$. Therefore, in this case, Lemma 41 implies that, with any budget n of size at least

$$\bar{c}2^{2\bar{k}}\left(\mathfrak{s}\mathrm{Log}\left(\frac{1}{\varepsilon}\right) + \mathrm{Log}\left(\frac{1}{\delta}\right)\right)\mathrm{Log}\left(\frac{d}{\varepsilon\delta}\right)\mathrm{Log}\left(\frac{1}{\varepsilon}\right) +$$
$$\bar{c}\sum_{k=\bar{k}}^{k_\varepsilon} \max\left\{\min\left\{a'2^{(3-k)\alpha/(1-\alpha)}, 1\right\}, \frac{\varepsilon}{\hat{\gamma}_\varepsilon}\right\}\frac{2^k}{\varepsilon}\left(d\mathrm{Log}\left(\frac{1}{\varepsilon}\right) + \mathrm{Log}\left(\frac{1}{\delta}\right)\right)\mathrm{Log}\left(\frac{d}{\varepsilon\delta}\right)\mathrm{Log}\left(\frac{1}{\hat{\gamma}_\varepsilon}\right), \tag{44}$$

Algorithm 1 produces a classifier $\hat{h}_n$ with $\mathrm{er}_{P_{XY}}(\hat{h}_n) - \mathrm{er}_{P_{XY}}(f^\star_{P_{XY}}) \le \varepsilon$ with probability at least $1-\delta$, and requests a number of labels at most n. This implies $\Lambda_{\mathrm{TN}(a,\alpha)}(\varepsilon,\delta)$ is at most (44).

Now note that

$$
\begin{aligned}
&2^{2\bar{k}}\left(\mathfrak{s}\mathrm{Log}\left(\frac{1}{\varepsilon}\right)+\mathrm{Log}\left(\frac{1}{\delta}\right)\right)\\
&\leq 256\left(\frac{k_\varepsilon a'}{8\varepsilon}\frac{d\mathrm{Log}\left(\frac{1}{\varepsilon}\right)+\mathrm{Log}\left(\frac{1}{\delta}\right)}{\mathfrak{s}\mathrm{Log}\left(\frac{1}{\varepsilon}\right)+\mathrm{Log}\left(\frac{1}{\delta}\right)}\right)^{2-2\alpha}\left(\mathfrak{s}\mathrm{Log}\left(\frac{1}{\varepsilon}\right)+\mathrm{Log}\left(\frac{1}{\delta}\right)\right)\\
&\leq 1024a^2\left(\frac{1}{\varepsilon}\right)^{2-2\alpha}\left(\frac{\mathfrak{s}\mathrm{Log}\left(\frac{1}{\varepsilon}\right)+\mathrm{Log}\left(\frac{1}{\delta}\right)}{d\mathrm{Log}\left(\frac{1}{\varepsilon}\right)+\mathrm{Log}\left(\frac{1}{\delta}\right)}\right)^{2\alpha-1}\left(d\mathrm{Log}\left(\frac{1}{\varepsilon}\right)+\mathrm{Log}\left(\frac{1}{\delta}\right)\right)\mathrm{Log}^{2-2\alpha}\left(\frac{1}{\varepsilon}\right).
\end{aligned}
$$

Also, since $\alpha \geq 1/2$, $2^{k\frac{1-2\alpha}{1-\alpha}}$ is nonincreasing in k, so that

$$
\begin{aligned}
&\sum_{k=\bar{k}}^{k_\varepsilon} a' 2^{(3-k)\alpha/(1-\alpha)}\frac{2^k}{\varepsilon}\left(d\mathrm{Log}\left(\frac{1}{\varepsilon}\right)+\mathrm{Log}\left(\frac{1}{\delta}\right)\right)\\
&\leq \frac{8a'k_\varepsilon}{\varepsilon}2^{(\bar{k}-3)\frac{1-2\alpha}{1-\alpha}}\left(d\mathrm{Log}\left(\frac{1}{\varepsilon}\right)+\mathrm{Log}\left(\frac{1}{\delta}\right)\right)\\
&\leq \frac{8a'k_\varepsilon}{\varepsilon}\left(\frac{k_\varepsilon a'}{8\varepsilon}\frac{d\mathrm{Log}\left(\frac{1}{\varepsilon}\right)+\mathrm{Log}\left(\frac{1}{\delta}\right)}{\mathfrak{s}\mathrm{Log}\left(\frac{1}{\varepsilon}\right)+\mathrm{Log}\left(\frac{1}{\delta}\right)}\right)^{1-2\alpha}\left(d\mathrm{Log}\left(\frac{1}{\varepsilon}\right)+\mathrm{Log}\left(\frac{1}{\delta}\right)\right)\\
&\leq 256a^2\left(\frac{1}{\varepsilon}\right)^{2-2\alpha}\left(\frac{\mathfrak{s}\mathrm{Log}\left(\frac{1}{\varepsilon}\right)+\mathrm{Log}\left(\frac{1}{\delta}\right)}{d\mathrm{Log}\left(\frac{1}{\varepsilon}\right)+\mathrm{Log}\left(\frac{1}{\delta}\right)}\right)^{2\alpha-1}\left(d\mathrm{Log}\left(\frac{1}{\varepsilon}\right)+\mathrm{Log}\left(\frac{1}{\delta}\right)\right)\mathrm{Log}^{2-2\alpha}\left(\frac{1}{\varepsilon}\right).
\end{aligned}
$$

Furthermore, by (43),

$$
\begin{aligned}
&\bar{c}\sum_{k=\bar{k}}^{k_\varepsilon}\frac{\varepsilon}{\hat{\gamma}_\varepsilon}\frac{2^k}{\varepsilon}\left(d\mathrm{Log}\left(\frac{1}{\varepsilon}\right)+\mathrm{Log}\left(\frac{1}{\delta}\right)\right)\leq 128a^2\left(\frac{1}{\varepsilon}\right)^{2-2\alpha}\left(d\mathrm{Log}\left(\frac{1}{\varepsilon}\right)+\mathrm{Log}\left(\frac{1}{\delta}\right)\right)\\
&\leq 128a^2\left(\frac{1}{\varepsilon}\right)^{2-2\alpha}\left(\frac{\mathfrak{s}\mathrm{Log}\left(\frac{1}{\varepsilon}\right)+\mathrm{Log}\left(\frac{1}{\delta}\right)}{d\mathrm{Log}\left(\frac{1}{\varepsilon}\right)+\mathrm{Log}\left(\frac{1}{\delta}\right)}\right)^{2\alpha-1}\left(d\mathrm{Log}\left(\frac{1}{\varepsilon}\right)+\mathrm{Log}\left(\frac{1}{\delta}\right)\right)\mathrm{Log}^{2-2\alpha}\left(\frac{1}{\varepsilon}\right).
\end{aligned}
$$

Therefore, since $\mathrm{Log}\left(\frac{1}{\hat{\gamma}_\varepsilon}\right)\leq \mathrm{Log}\left(\frac{2}{\varepsilon}\right)\leq 2\mathrm{Log}\left(\frac{1}{\varepsilon}\right)$, (44) is at most

$$
2^{11}\bar{c}a^2\left(\frac{1}{\varepsilon}\right)^{2-2\alpha}\left(\frac{\mathfrak{s}\mathrm{Log}\left(\frac{1}{\varepsilon}\right)+\mathrm{Log}\left(\frac{1}{\delta}\right)}{d\mathrm{Log}\left(\frac{1}{\varepsilon}\right)+\mathrm{Log}\left(\frac{1}{\delta}\right)}\right)^{2\alpha-1}\left(d\mathrm{Log}\left(\frac{1}{\varepsilon}\right)+\mathrm{Log}\left(\frac{1}{\delta}\right)\right)\mathrm{Log}\left(\frac{d}{\varepsilon\delta}\right)\mathrm{Log}^{3-2\alpha}\left(\frac{1}{\varepsilon}\right). \tag{45}
$$

The upper bound for the case $\alpha \geq 1/2$ and $\frac{\mathfrak{s}}{d} < \frac{1}{a^{1/\alpha}\varepsilon}$ then follows by further relaxing this (purely to simplify the theorem statement), noting that $\mathrm{Log}^{3-2\alpha}\left(\frac{1}{\varepsilon}\right)\leq \mathrm{Log}^2\left(\frac{1}{\varepsilon}\right)$, and $\frac{\mathfrak{s}\mathrm{Log}\left(\frac{1}{\varepsilon}\right)+\mathrm{Log}\left(\frac{1}{\delta}\right)}{d\mathrm{Log}\left(\frac{1}{\varepsilon}\right)+\mathrm{Log}\left(\frac{1}{\delta}\right)}\leq \frac{\mathfrak{s}}{d}$.

Next, we turn to establishing the lower bound. Fix any $a \in [4,\infty)$, $\alpha \in (0,1)$, $\delta \in (0,1/24]$, and $\varepsilon \in \left(0, 1/(24a^{1/\alpha})\right)$. For this range of values, the recent article of Hanneke (2014) proves a lower bound of

$$
\Lambda_{\mathrm{TN}(a,\alpha)}(\varepsilon,\delta) \gtrsim a^2\left(\frac{1}{\varepsilon}\right)^{2-2\alpha}\left(d+\mathrm{Log}\left(\frac{1}{\delta}\right)\right),
$$

based on techniques of Kääriäinen (2006); Beygelzimer, Dasgupta, and Langford (2009); Hanneke (2011). It remains only to establish the remaining term in the lower bound for the case when $\alpha > 1/2$, via Lemma 26. In the cases that $\mathfrak{s} \leq 2$, this term is implied by the above $a^2\varepsilon^{2\alpha-2}\mathrm{Log}\left(\frac{1}{\delta}\right)$ lower bound. For the remainder of the proof, suppose $\mathfrak{s} \geq 3$ and $\alpha > 1/2$. Let

$$k = \min\left\{\mathfrak{s}-1, \left\lfloor \frac{(a')^{\frac{\alpha-1}{\alpha}}}{\varepsilon} \right\rfloor, \left\lfloor \frac{a'}{\varepsilon} 4^{-\frac{1}{1-\alpha}} \right\rfloor \right\},$$

$\beta = \frac{1}{2} - \left(\frac{k\varepsilon}{a'}\right)^{1-\alpha}$, and $\zeta = \frac{2\varepsilon}{1-2\beta}$; note that $\zeta \in (0,1]$, $\beta \in [0, 1/2)$, and $2 \leq k \leq \min\{\mathfrak{s}-1, \lfloor 1/\zeta \rfloor\}$; in particular, the fact that $k \leq \lfloor 1/\zeta \rfloor$ is established by concavity of the $x \mapsto \frac{(a')^{\alpha-1}}{\varepsilon^\alpha} x^{1-\alpha}$ function, which equals x at both $x = 0$ and $x = x_0 = \frac{(a')^{\frac{\alpha-1}{\alpha}}}{\varepsilon}$; since this function is $1/\zeta$ at $x = k$, and $0 < k \leq x_0$, concavity of the function implies $1/\zeta \geq k$, and integrality of k implies $\lfloor 1/\zeta \rfloor \geq k$ as well. Also note that any $\mathcal{P}_{XY} \in \mathrm{RR}(k, \zeta, \beta)$ has a marginal distribution $\mathcal{P}$ such that

$$\begin{aligned}\mathcal{P}\left(x : |\eta(x;\mathcal{P}_{XY}) - 1/2| \leq 1/2 - \beta\right) &= k\zeta = k\varepsilon\frac{2}{1-2\beta} \\ &= a'\left(1/2-\beta\right)^{\frac{1}{1-\alpha}}\frac{2}{1-2\beta} = a'\left(1/2-\beta\right)^{\frac{\alpha}{1-\alpha}}.\end{aligned}$$

Since every point x in the support of $\mathcal{P}_{k,\zeta}$ has either $|\eta(x;\mathcal{P}_{XY}) - 1/2| = 1/2 - \beta$ or $|\eta(x;\mathcal{P}_{XY}) - 1/2| = 1/2$, this implies that any $\gamma \in [1/2-\beta, 1/2)$ has $\mathcal{P}(x : |\eta(x;\mathcal{P}_{XY}) - 1/2| \leq \gamma) = \mathcal{P}\left(x : |\eta(x;\mathcal{P}_{XY}) - 1/2| \leq 1/2 - \beta\right) = a'\left(1/2-\beta\right)^{\alpha/(1-\alpha)} \leq a'\gamma^{\alpha/(1-\alpha)}$, while any $\gamma \geq 1/2$ always has $\mathcal{P}\left(x : |\eta(x;\mathcal{P}_{XY}) - 1/2| \leq \gamma\right) = 1 \leq a'\gamma^{\alpha/(1-\alpha)}$. Furthermore, any $\gamma \in (0, 1/2-\beta)$ has $\mathcal{P}(x : |\eta(x;\mathcal{P}_{XY}) - 1/2| \leq \gamma) = 0 \leq a'\gamma^{\alpha/(1-\alpha)}$. Thus, $\mathcal{P}_{XY} \in \mathrm{TN}(a, \alpha)$ as well. Since this holds for every $\mathcal{P}_{XY} \in \mathrm{RR}(k, \zeta, \beta)$, this implies $\mathrm{RR}(k, \zeta, \beta) \subseteq \mathrm{TN}(a, \alpha)$. Therefore, Lemma 26 implies

$$\begin{aligned}\Lambda_{\mathrm{TN}(a,\alpha)}(\varepsilon, \delta) \geq \Lambda_{\mathrm{RR}(k,\zeta,\beta)}(\varepsilon, \delta) &= \Lambda_{\mathrm{RR}(k,\zeta,\beta)}((\zeta/2)(1-2\beta), \delta) \\ &\geq \frac{\beta(k-1)\ln\left(\frac{1}{4\delta}\right)}{3(1-2\beta)^2} \gtrsim \frac{\beta(k-1)}{(1-2\beta)^2}\mathrm{Log}\left(\frac{1}{\delta}\right). \qquad (46)\end{aligned}$$

Finally, note that

$$\begin{aligned}&\frac{\beta(k-1)}{(1-2\beta)^2} = \left(\frac{1}{2} - \left(\frac{k\varepsilon}{a'}\right)^{1-\alpha}\right)\frac{1}{4}\left(\frac{a'}{k\varepsilon}\right)^{2-2\alpha}(k-1) \geq \frac{1}{16}\left(\frac{a'}{\varepsilon}\right)^{2-2\alpha} k^{2\alpha-2}(k-1) \\ &\geq \frac{1}{32}\left(\frac{a'}{\varepsilon}\right)^{2-2\alpha}(k-1)^{2\alpha-1} \geq \frac{a^2}{64}\left(\frac{1}{\varepsilon}\right)^{2-2\alpha}(k-1)^{2\alpha-1}. \qquad (47)\end{aligned}$$

Since $a \geq 4$,

$$\begin{aligned}\left(a'\right)^{\frac{\alpha-1}{\alpha}} &= a'\left(a'\right)^{-1/\alpha} = a'(1-\alpha)^{-1/\alpha}(2\alpha)^{-1/(1-\alpha)}a^{-\frac{1}{\alpha(1-\alpha)}} \\ &\leq a'(1-\alpha)^{-1/\alpha}(2\alpha)^{-1/(1-\alpha)}4^{-\frac{1}{\alpha(1-\alpha)}} = a'\left(4^{1/\alpha}(1-\alpha)^{(1-\alpha)/\alpha}(2\alpha)\right)^{-1/(1-\alpha)}.\end{aligned}$$

One can easily verify that $4^{1/\alpha}(1-\alpha)^{(1-\alpha)/\alpha}(2\alpha) \geq 6$ for $\alpha \in (1/2, 1)$ (with minimum achieved at $\alpha = 3/4$), so that $a'\left(4^{1/\alpha}(1-\alpha)^{(1-\alpha)/\alpha}(2\alpha)\right)^{-1/(1-\alpha)} \leq a' 6^{-1/(1-\alpha)} \leq a' 4^{-1/(1-\alpha)}$. Thus, $\frac{(a')^{\frac{\alpha-1}{\alpha}}}{\varepsilon} \leq \frac{a'}{\varepsilon} 4^{-\frac{1}{1-\alpha}}$, so that the third term in the definition of k is redundant. Therefore, (47) is at least

$$\frac{a^2}{64}\left(\frac{1}{\varepsilon}\right)^{2-2\alpha} \min\left\{\mathfrak{s}-2, \frac{(a')^{\frac{\alpha-1}{\alpha}}}{\varepsilon} - 2\right\}^{2\alpha-1} \geq \frac{a^2}{64}\left(\frac{1}{\varepsilon}\right)^{2-2\alpha} \min\left\{\mathfrak{s}-2, \frac{1}{2a^{1/\alpha}\varepsilon} - 2\right\}^{2\alpha-1}$$
$$\geq \frac{a^2}{64}\left(\frac{1}{\varepsilon}\right)^{2-2\alpha} \min\left\{\frac{\mathfrak{s}}{3}, \frac{1}{3a^{1/\alpha}\varepsilon}\right\}^{2\alpha-1} \geq \frac{a^2}{192}\left(\frac{1}{\varepsilon}\right)^{2-2\alpha} \min\left\{\mathfrak{s}, \frac{1}{a^{1/\alpha}\varepsilon}\right\}^{2\alpha-1}.$$

Plugging this into (46) completes the proof. ■

As an aside, we note that it is possible to improve the logarithmic factors in the upper bound in Theorem 5. One clear refinement comes from using (45) directly (rather than relaxing the factor depending on $\mathfrak{s}$). We can further reduce the bound by another logarithmic factor when α is bounded away from $1/2$ by noting that the summations of terms $2^{(k-3)\frac{1-2\alpha}{1-\alpha}}$ in the above proof are geometric in that case. We also note that, for very large values of a, the bounds (proven below) for $\Lambda_{\mathrm{BE}(1/2)}(\varepsilon, \delta)$ may be more informative than those derived above.

Proof of Theorem 6 The technique leading to Lemma 41 does not apply to $\mathrm{BC}(a, \alpha)$, since we are not guaranteed $f^\star_{\mathcal{P}_{XY}} \in \mathbb{C}$ for $\mathcal{P}_{XY} \in \mathrm{BC}(a, \alpha)$. We therefore base the upper bounds in Theorem 6 directly on existing results in the literature, in combination with Theorem 10. Thus, the proof of this upper bound does not provide any new insights on improving the design of active learning algorithms for distributions in $\mathrm{BC}(a, \alpha)$. Rather, it merely re-expresses the known results, in terms of the star number instead of a distribution-dependent complexity measure. The lower bounds are directly inherited from Theorem 5.

Fix any $a \in [1, \infty)$, $\alpha \in [0, 1]$, and $\varepsilon, \delta \in (0, 1)$. Following the work of Hanneke (2009a, 2011) and Koltchinskii (2010), the recent work of Hanneke and Yang (2012) studies an algorithm proposed by Hanneke (2012) (a modified variant of the A^2 algorithm of Balcan, Beygelzimer, and Langford, 2006, 2009), and shows that there exists a finite universal constant $\mathring{c} \geq 1$ such that, for any $\mathcal{P}_{XY} \in \mathrm{BC}(a, \alpha)$, for any budget n of size at least

$$\mathring{c}a^2\left(\frac{1}{\varepsilon}\right)^{2-2\alpha} \theta_{\mathcal{P}_{XY}}\left(a\varepsilon^\alpha\right)\left(d\mathrm{Log}\left(\theta_{\mathcal{P}_{XY}}\left(a\varepsilon^\alpha\right)\right) + \mathrm{Log}\left(\frac{\mathrm{Log}(1/\varepsilon)}{\delta}\right)\right)\mathrm{Log}\left(\frac{1}{\varepsilon}\right), \tag{48}$$

the algorithm produces a classifier $\hat{h}_n$ with $\mathrm{er}_{\mathcal{P}_{XY}}(\hat{h}_n) - \inf_{h\in\mathbb{C}} \mathrm{er}_{\mathcal{P}_{XY}}(h) \leq \varepsilon$ with probability at least $1-\delta/4$, and requests a number of labels at most n (see also Hanneke, 2009b,a, 2011, 2012, 2014; Koltchinskii, 2010, for similar results for related methods). By Theorem 10, when $a\varepsilon^\alpha \leq 1$, (48) is at most

$$\mathring{c}a^2\left(\frac{1}{\varepsilon}\right)^{2-2\alpha} \min\left\{\mathfrak{s}, \frac{1}{a\varepsilon^\alpha}\right\}\left(d\mathrm{Log}\left(\min\left\{\mathfrak{s}, \frac{1}{a\varepsilon^\alpha}\right\}\right) + \mathrm{Log}\left(\frac{\mathrm{Log}(1/\varepsilon)}{\delta}\right)\right)\mathrm{Log}\left(\frac{1}{\varepsilon}\right), \tag{49}$$

which is therefore an upper bound on $\Lambda_{\mathrm{BC}(a,\alpha)}(\varepsilon, \delta)$. We can also extend this to the case $a\varepsilon^\alpha > 1$ as follows. Vapnik and Chervonenkis (1971); Vapnik (1982, 1998) have proven that

the sample complexity of passive learning satisfies

$$\mathcal{M}_{\mathrm{AG}(1)}(\varepsilon,\delta) \lesssim \frac{1}{\varepsilon^2}\left(d\mathrm{Log}\left(\frac{1}{\varepsilon}\right)+\mathrm{Log}\left(\frac{1}{\delta}\right)\right).$$

In the case $a\varepsilon^\alpha > 1$, this is at most

$$\begin{aligned}
&a\left(\frac{1}{\varepsilon}\right)^{2-\alpha}\left(d\mathrm{Log}\left(\frac{1}{\varepsilon}\right)+\mathrm{Log}\left(\frac{1}{\delta}\right)\right)\\
&= a^2\left(\frac{1}{\varepsilon}\right)^{2-2\alpha}\min\left\{\mathfrak{s},\frac{1}{a\varepsilon^\alpha}\right\}\left(d\mathrm{Log}\left(\frac{1}{\varepsilon}\right)+\mathrm{Log}\left(\frac{1}{\delta}\right)\right)\\
&\le a^2\left(\frac{1}{\varepsilon}\right)^{2-2\alpha}\min\left\{\mathfrak{s},\frac{1}{a\varepsilon^\alpha}\right\}\left(d\mathrm{Log}\left(\min\left\{\mathfrak{s},\frac{1}{a\varepsilon^\alpha}\right\}\right)+\mathrm{Log}\left(\frac{\mathrm{Log}(1/\varepsilon)}{\delta}\right)\right)\mathrm{Log}\left(\frac{1}{\varepsilon}\right).
\end{aligned}$$

Therefore, since $\Lambda_{\mathrm{AG}(1)}(\varepsilon,\delta) \le \mathcal{M}_{\mathrm{AG}(1)}(\varepsilon,\delta)$ and $\mathrm{BC}(a,\alpha) \subseteq \mathrm{AG}(1)$, we may conclude that, regardless of whether $a\varepsilon^\alpha$ is greater than or less than 1, we have that $\Lambda_{\mathrm{BC}(a,\alpha)}(\varepsilon,\delta)$ is bounded by a value proportional to (49). To match the form of the upper bound stated in Theorem 6, we can simply relax this, noting that $d\mathrm{Log}\left(\min\left\{\mathfrak{s},\frac{1}{a\varepsilon^\alpha}\right\}\right)+\mathrm{Log}\left(\frac{\mathrm{Log}(1/\varepsilon)}{\delta}\right) \le 2d\mathrm{Log}\left(\frac{1}{\varepsilon}\right)+\mathrm{Log}\left(\frac{1}{\delta}\right) \le 2d\mathrm{Log}\left(\frac{1}{\varepsilon\delta}\right)$.

Next, turning to the lower bound, recall that $\mathrm{TN}(a,\alpha) \subseteq \mathrm{BC}(a,\alpha)$, so that $\Lambda_{\mathrm{TN}(a,\alpha)}(\varepsilon,\delta) \le \Lambda_{\mathrm{BC}(a,\alpha)}(\varepsilon,\delta)$ (Mammen and Tsybakov, 1999; Tsybakov, 2004). Thus, the lower bound in Theorem 5 (proven above) for $\Lambda_{\mathrm{TN}(a,\alpha)}(\varepsilon,\delta)$ also applies to $\Lambda_{\mathrm{BC}(a,\alpha)}(\varepsilon,\delta)$. ■

Proof of Theorem 7 Again, we begin with the upper bound. Fix any $\nu \in [0,1/2]$, $\varepsilon,\delta \in (0,1)$, and $\mathcal{P}_{XY} \in \mathrm{BE}(\nu)$. The case $\nu = 0$ is already addressed by the upper bound in Theorem 3; we therefore focus the remainder of the proof on the case of $\nu > 0$. For $(X,Y) \sim \mathcal{P}_{XY}$, any $x \in \mathcal{X}$ has $1-2\mathbb{P}(Y \ne f^\star_{\mathcal{P}_{XY}}(X)|X=x) = 2\gamma_x$. Therefore, for any $\gamma \in [0,1/2)$, any $x \in \mathcal{X}$ with $\gamma_x \le \gamma$ has $\mathbb{P}(Y \ne f^\star_{\mathcal{P}_{XY}}(X)|X=x) \ge 1/2-\gamma$. Thus, Markov's inequality implies

$$\mathcal{P}(x:\gamma_x \le \gamma) \le \mathcal{P}(x:\mathbb{P}(Y \ne f^\star_{\mathcal{P}_{XY}}(X)|X=x) \ge 1/2-\gamma) \le \frac{2}{1-2\gamma}\mathrm{er}_{\mathcal{P}_{XY}}(f^\star_{\mathcal{P}_{XY}}) \le \frac{2\nu}{1-2\gamma}. \tag{50}$$

In particular, this implies that for $\gamma \le \frac{\varepsilon}{4\nu+2\varepsilon}$, $\gamma\mathcal{P}(x:\gamma_x \le \gamma) \le \frac{2\nu\gamma}{1-2\gamma} \le \frac{2\nu/(2\nu+\varepsilon)}{1-\varepsilon/(2\nu+\varepsilon)}\frac{\varepsilon}{2} = \frac{\varepsilon}{2}$. Thus, $\gamma_\varepsilon \ge \frac{\varepsilon}{4\nu+2\varepsilon}$. We can therefore take $\hat{\gamma}_\varepsilon = \max\left\{\frac{\varepsilon}{4\nu+2\varepsilon},\frac{\varepsilon}{2}\right\}$.

Also note that any $\gamma \ge 0$ has $\mathcal{P}(x:\gamma_x \le \gamma) \le 1$, so that together with (50), we have $\mathcal{P}(x:\gamma_x \le \gamma) \le \frac{2\nu}{1-\min\{2\gamma,1-2\nu\}}$. Now taking $\bar{k}=2$, Lemma 41 implies that, with any budget n of size at least

$$\bar{c}\sum_{k=2}^{k_\varepsilon}\max\left\{\frac{2\nu}{1-\min\left\{2^{4-k},1-2\nu\right\}},\frac{\varepsilon}{\hat{\gamma}_\varepsilon}\right\}\frac{2^k}{\varepsilon}\left(d\mathrm{Log}\left(\frac{1}{\varepsilon}\right)+\mathrm{Log}\left(\frac{1}{\delta}\right)\right)\mathrm{Log}\left(\frac{d}{\varepsilon\delta}\right)\mathrm{Log}\left(\frac{1}{\hat{\gamma}_\varepsilon}\right), \tag{51}$$

Algorithm 1 produces a classifier $\hat{h}_n$ with $\mathrm{er}_{\mathcal{P}_{XY}}(\hat{h}_n) - \mathrm{er}_{\mathcal{P}_{XY}}(f^\star_{\mathcal{P}_{XY}}) \le \varepsilon$ with probability at least $1-\delta$, and requests a number of labels at most n. This implies $\Lambda_{\mathrm{BE}(\nu)}(\varepsilon,\delta)$ is at most

(51). Now note that

$$\sum_{k=2}^{k_\varepsilon} \frac{\varepsilon}{\hat{\gamma}_\varepsilon} \frac{2^k}{\varepsilon} \leq \frac{1}{\hat{\gamma}_\varepsilon} 2^{1+k_\varepsilon} \leq 512 \left(\frac{\nu+\varepsilon}{\varepsilon}\right)^2. \tag{52}$$

Next, we have

$$\begin{aligned}
&\sum_{k=2}^{k_\varepsilon} \frac{2\nu}{1-\min\left\{2^{4-k}, 1-2\nu\right\}} \frac{2^k}{\varepsilon} \leq \frac{28}{\varepsilon} + \sum_{k=5}^{k_\varepsilon} \frac{2\nu}{1-2^{4-k}} \frac{2^k}{\varepsilon} \leq \frac{28}{\varepsilon} + \sum_{k=5}^{k_\varepsilon} \frac{4\nu}{\varepsilon} 2^k \\
&\leq \frac{28}{\varepsilon} + \frac{4\nu}{\varepsilon} 2^{1+k_\varepsilon} \leq \frac{28}{\varepsilon} + \frac{128\nu}{\varepsilon\hat{\gamma}_\varepsilon} \leq \frac{28}{\varepsilon} + 512 \left(\frac{\nu+\varepsilon}{\varepsilon}\right)^2.
\end{aligned}$$

Therefore, (51) is at most

$$\begin{aligned}
&2^{10}\bar{c}\left(\left(\frac{\nu+\varepsilon}{\varepsilon}\right)^2 + \frac{1}{\varepsilon}\right)\left(d\mathrm{Log}\left(\frac{1}{\varepsilon}\right) + \mathrm{Log}\left(\frac{1}{\delta}\right)\right)\mathrm{Log}\left(\frac{d}{\varepsilon\delta}\right)\mathrm{Log}\left(\frac{1}{\hat{\gamma}_\varepsilon}\right) \\
&\quad \leq 2^{10}3\bar{c}\left(\left(\frac{\nu+\varepsilon}{\varepsilon}\right)^2 + \frac{1}{\varepsilon}\right)\left(d\mathrm{Log}\left(\frac{1}{\varepsilon}\right) + \mathrm{Log}\left(\frac{1}{\delta}\right)\right)\mathrm{Log}\left(\frac{d}{\varepsilon\delta}\right)\mathrm{Log}\left(\frac{\nu+\varepsilon}{\varepsilon}\right). \quad (53)
\end{aligned}$$

Next, consider taking $\bar{k} = 5$. Lemma 41 implies that, with any budget n of size at least

$$\begin{aligned}
&2^{10}\bar{c}\left(\mathfrak{s}\mathrm{Log}\left(\frac{1}{\varepsilon}\right) + \mathrm{Log}\left(\frac{1}{\delta}\right)\right)\mathrm{Log}\left(\frac{d}{\varepsilon\delta}\right)\mathrm{Log}\left(\frac{1}{\varepsilon}\right) \\
&\quad + \bar{c}\sum_{k=5}^{k_\varepsilon} \max\left\{\frac{2\nu}{1-2^{4-k}}, \frac{\varepsilon}{\hat{\gamma}_\varepsilon}\right\} \frac{2^k}{\varepsilon}\left(d\mathrm{Log}\left(\frac{1}{\varepsilon}\right) + \mathrm{Log}\left(\frac{1}{\delta}\right)\right)\mathrm{Log}\left(\frac{d}{\varepsilon\delta}\right)\mathrm{Log}\left(\frac{1}{\hat{\gamma}_\varepsilon}\right), \quad (54)
\end{aligned}$$

Algorithm 1 produces a classifier $\hat{h}_n$ with $\mathrm{er}_{\mathcal{P}_{XY}}(\hat{h}_n) - \mathrm{er}_{\mathcal{P}_{XY}}(f^\star_{\mathcal{P}_{XY}}) \leq \varepsilon$ with probability at least $1-\delta$, and requests a number of labels at most n. This implies $\Lambda_{\mathrm{BE}(\nu)}(\varepsilon, \delta)$ is at most (54). As above, we have

$$\sum_{k=5}^{k_\varepsilon} \frac{2\nu}{1-2^{4-k}} \frac{2^k}{\varepsilon} \leq 512 \left(\frac{\nu+\varepsilon}{\varepsilon}\right)^2.$$

Combined with (52), this implies (54) is at most

$$\begin{aligned}
&2^{10}\bar{c}\left(\mathfrak{s}\mathrm{Log}\left(\frac{1}{\varepsilon}\right) + \mathrm{Log}\left(\frac{1}{\delta}\right)\right)\mathrm{Log}\left(\frac{d}{\varepsilon\delta}\right)\mathrm{Log}\left(\frac{1}{\varepsilon}\right) \\
&+ 2^{10}\bar{c}\left(\frac{\nu+\varepsilon}{\varepsilon}\right)^2\left(d\mathrm{Log}\left(\frac{1}{\varepsilon}\right) + \mathrm{Log}\left(\frac{1}{\delta}\right)\right)\mathrm{Log}\left(\frac{d}{\varepsilon\delta}\right)\mathrm{Log}\left(\frac{1}{\hat{\gamma}_\varepsilon}\right) \\
&\leq 2^{10}\bar{c}\left(\mathfrak{s}\mathrm{Log}\left(\frac{1}{\varepsilon}\right) + \mathrm{Log}\left(\frac{1}{\delta}\right)\right)\mathrm{Log}\left(\frac{d}{\varepsilon\delta}\right)\mathrm{Log}\left(\frac{1}{\varepsilon}\right) \\
&+ 2^{10}3\bar{c}\left(\frac{\nu+\varepsilon}{\varepsilon}\right)^2\left(d\mathrm{Log}\left(\frac{1}{\varepsilon}\right) + \mathrm{Log}\left(\frac{1}{\delta}\right)\right)\mathrm{Log}\left(\frac{d}{\varepsilon\delta}\right)\mathrm{Log}\left(\frac{\nu+\varepsilon}{\varepsilon}\right). \quad (55)
\end{aligned}$$

In particular, when $\left(\mathfrak{s}\mathrm{Log}\left(\frac{1}{\varepsilon}\right) + \mathrm{Log}\left(\frac{1}{\delta}\right)\right)\mathrm{Log}\left(\frac{1}{\varepsilon}\right) < \frac{3}{\varepsilon}\left(d\mathrm{Log}\left(\frac{1}{\varepsilon}\right) + \mathrm{Log}\left(\frac{1}{\delta}\right)\right)\mathrm{Log}\left(\frac{\nu+\varepsilon}{\varepsilon}\right)$, this is smaller than (53). Thus, the minimum of these two expressions upper bounds $\Lambda_{\mathrm{BE}(\nu)}(\varepsilon, \delta)$.

To simplify the expression of this bound into the form given in the statement of Theorem 7, we note that $d\mathrm{Log}\left(\frac{1}{\varepsilon}\right) + \mathrm{Log}\left(\frac{1}{\delta}\right) \leq d\mathrm{Log}\left(\frac{1}{\varepsilon\delta}\right)$, $\mathfrak{s}\mathrm{Log}\left(\frac{1}{\varepsilon}\right) + \mathrm{Log}\left(\frac{1}{\delta}\right) \leq \mathfrak{s}\mathrm{Log}\left(\frac{1}{\varepsilon\delta}\right)$, $\mathrm{Log}\left(\frac{\nu+\varepsilon}{\varepsilon}\right) \leq \mathrm{Log}\left(\frac{1}{\varepsilon}\right)$,$\left(\frac{\nu+\varepsilon}{\varepsilon}\right)^2 \leq 4\frac{\max\{\nu,\varepsilon\}^2}{\varepsilon^2} \leq 4\left(\frac{\nu^2}{\varepsilon^2}+1\right)$, and $d \leq \min\left\{\mathfrak{s}, \frac{d}{\varepsilon}\right\}$, so that the minimum of (53) and (55) is at most

$$2^{12}3\bar{c}\left(\left(\frac{\nu^2}{\varepsilon^2}+1\right)d + \min\left\{\mathfrak{s}, \frac{d}{\varepsilon}\right\}\right)\mathrm{Log}\left(\frac{d}{\varepsilon\delta}\right)\mathrm{Log}\left(\frac{1}{\varepsilon\delta}\right)\mathrm{Log}\left(\frac{1}{\varepsilon}\right)$$
$$\leq 2^{13}3\bar{c}\left(\frac{\nu^2}{\varepsilon^2}d + \min\left\{\mathfrak{s}, \frac{d}{\varepsilon}\right\}\right)\mathrm{Log}\left(\frac{d}{\varepsilon\delta}\right)\mathrm{Log}\left(\frac{1}{\varepsilon\delta}\right)\mathrm{Log}\left(\frac{1}{\varepsilon}\right).$$

This completes the proof of the upper bound.

Next, we turn to establishing the lower bound. Fix $\nu \in [0, 1/2)$, $\varepsilon \in (0, (1-2\nu)/24)$, and $\delta \in (0, 1/24]$. Based on the works of Kääriäinen (2006); Hanneke (2007a); Beygelzimer, Dasgupta, and Langford (2009), the recent article of Hanneke (2014) contains the following lower bound (in the proof of Theorem 4.3 there), letting $\gamma = \frac{12\varepsilon}{\nu+12\varepsilon}$.

$$\Lambda_{\mathrm{BE}(\nu)}(\varepsilon,\delta) \geq \max\left\{2\left\lfloor\frac{1-\gamma^2}{2\gamma^2}\ln\left(\frac{1}{8\delta(1-2\delta)}\right)\right\rfloor, \frac{d-1}{6}\left\lfloor\frac{1-\gamma^2}{2\gamma^2}\ln\left(\frac{9}{8}\right)\right\rfloor\right\}$$
$$\geq \max\left\{2\left\lfloor\frac{1-\gamma^2}{2\gamma^2}\ln\left(\frac{1}{8\delta}\right)\right\rfloor, \frac{d-1}{6}\left\lfloor\frac{1-\gamma^2}{17\gamma^2}\right\rfloor\right\} \tag{56}$$

If $\frac{1-\gamma^2}{2\gamma^2}\ln\left(\frac{1}{8\delta}\right) \geq 1$, then $2\left\lfloor\frac{1-\gamma^2}{2\gamma^2}\ln\left(\frac{1}{8\delta}\right)\right\rfloor \geq \frac{1-\gamma^2}{2\gamma^2}\ln\left(\frac{1}{8\delta}\right)$, so that (56) implies $\Lambda_{\mathrm{BE}(\nu)}(\varepsilon,\delta) \gtrsim \frac{1-\gamma^2}{\gamma^2}\mathrm{Log}\left(\frac{1}{\delta}\right)$. Otherwise, if $\frac{1-\gamma^2}{2\gamma^2}\ln\left(\frac{1}{8\delta}\right) < 1$, then since $\mathrm{RE} \subseteq \mathrm{BE}(\nu)$, and $|\mathbb{C}| \geq 2$ implies $d \geq 1 > \frac{1-\gamma^2}{2\gamma^2}\ln\left(\frac{1}{8\delta}\right)$, Theorem 3 (proven above) implies $\Lambda_{\mathrm{BE}(\nu)}(\varepsilon,\delta) \geq \Lambda_{\mathrm{RE}}(\varepsilon,\delta) \gtrsim d \gtrsim \frac{1-\gamma^2}{\gamma^2}\mathrm{Log}\left(\frac{1}{\delta}\right)$ in this case as well. If $d = 1$, these observations further imply $\Lambda_{\mathrm{BE}(\nu)}(\varepsilon,\delta) \gtrsim d\frac{1-\gamma^2}{\gamma^2}$. On the other hand, if $d \geq 2$, and if $\frac{1-\gamma^2}{17\gamma^2} \geq 1$, then $\frac{d-1}{6}\left\lfloor\frac{1-\gamma^2}{17\gamma^2}\right\rfloor \geq \frac{d}{408}\frac{1-\gamma^2}{\gamma^2}$, so that (56) implies $\Lambda_{\mathrm{BE}(\nu)}(\varepsilon,\delta) \gtrsim d\frac{1-\gamma^2}{\gamma^2}$. Otherwise, if $\frac{1-\gamma^2}{17\gamma^2} < 1$, then since $\mathrm{RE} \subseteq \mathrm{BE}(\nu)$, Theorem 3 implies we still have $\Lambda_{\mathrm{BE}(\nu)}(\varepsilon,\delta) \geq \Lambda_{\mathrm{RE}}(\varepsilon,\delta) \gtrsim d \gtrsim d\frac{1-\gamma^2}{\gamma^2}$ in this case as well. Altogether, we have that

$$\Lambda_{\mathrm{BE}(\nu)}(\varepsilon,\delta) \gtrsim \frac{1-\gamma^2}{\gamma^2}\max\left\{d, \mathrm{Log}\left(\frac{1}{\delta}\right)\right\} \gtrsim \frac{1-\gamma^2}{\gamma^2}\left(d + \mathrm{Log}\left(\frac{1}{\delta}\right)\right). \tag{57}$$

When $\nu \geq 12\varepsilon$, $\gamma \leq 1/2$, so that (57) implies

$$\Lambda_{\mathrm{BE}(\nu)}(\varepsilon,\delta) \gtrsim \frac{1}{\gamma^2}\left(d + \mathrm{Log}\left(\frac{1}{\delta}\right)\right) = \left(\frac{\nu+12\varepsilon}{12\varepsilon}\right)^2\left(d + \mathrm{Log}\left(\frac{1}{\delta}\right)\right) \gtrsim \frac{\nu^2}{\varepsilon^2}\left(d + \mathrm{Log}\left(\frac{1}{\delta}\right)\right).$$

Otherwise, if $\nu < 12\varepsilon$, then

$$\frac{1-\gamma^2}{\gamma^2} = \frac{(1-\gamma)(1+\gamma)}{\gamma^2} = \left(\frac{\nu+12\varepsilon}{12\varepsilon}\right)^2\left(\frac{\nu}{\nu+12\varepsilon}\right)\left(\frac{\nu+24\varepsilon}{\nu+12\varepsilon}\right) \geq \frac{\nu}{\nu+12\varepsilon} \geq \frac{\nu}{12\varepsilon} \geq \frac{\nu^2}{144\varepsilon^2}. \tag{58}$$

Therefore, if $\nu < 12\varepsilon$, (57) implies that $\Lambda_{\mathrm{BE}(\nu)}(\varepsilon,\delta) \gtrsim \frac{\nu^2}{\varepsilon^2}\left(d + \mathrm{Log}\left(\frac{1}{\delta}\right)\right)$ in this case as well. It remains only to establish the final term in the lower bound. For this, we simply note that $\mathrm{RE} \subseteq \mathrm{BE}(\nu)$, so that Theorem 3 implies $\Lambda_{\mathrm{BE}(\nu)}(\varepsilon,\delta) \geq \Lambda_{\mathrm{RE}}(\varepsilon,\delta) \gtrsim \min\left\{\mathfrak{s}, \frac{1}{\varepsilon}\right\}$. Combining these results implies

$$\Lambda_{\mathrm{BE}(\nu)}(\varepsilon,\delta) \gtrsim \max\left\{\frac{\nu^2}{\varepsilon^2}\left(d + \mathrm{Log}\left(\frac{1}{\delta}\right)\right), \min\left\{\mathfrak{s}, \frac{1}{\varepsilon}\right\}\right\} \gtrsim \frac{\nu^2}{\varepsilon^2}\left(d + \mathrm{Log}\left(\frac{1}{\delta}\right)\right) + \min\left\{\mathfrak{s}, \frac{1}{\varepsilon}\right\}.$$

■

Examining the proof of the lower bound for $\Lambda_{\mathrm{BE}(\nu)}(\varepsilon,\delta)$, we note that this argument also establishes a slightly stronger lower bound in the case $\varepsilon > \nu$. Specifically, if we use the expression just left of the right-most inequality in (58), rather than the right-most expression, we find that we can add a term $\frac{\nu}{\varepsilon}\mathrm{Log}\left(\frac{1}{\delta}\right)$ to the stated lower bound. This term can be larger than the stated term $\frac{\nu^2}{\varepsilon^2}\mathrm{Log}\left(\frac{1}{\delta}\right)$ when $\varepsilon > \nu$. Additionally, since $\mathrm{RE} \subseteq \mathrm{BE}(\nu)$, we can of course also add a term d to the stated lower bound, which again would increase the bound when $\varepsilon > \nu$.

Proof of Theorem 8 Again, we begin with the upper bounds. As with the proof of Theorem 6, we cannot use the technique leading to Lemma 41; we turn instead to a simple combination of an upper bound from the literature, combined with Theorem 10.

Fix any $\nu \in [0,1]$ and $\varepsilon, \delta \in (0,1)$. Following the work of Hanneke (2007b); Dasgupta, Hsu, and Monteleoni (2007); Koltchinskii (2010), the recent work of Hanneke (2014) studies a modified variant of the A^2 algorithm of Balcan, Beygelzimer, and Langford (2006, 2009), showing that there exists a finite universal constant $\ddot{c} \geq 1$ such that, for any $\mathcal{P}_{XY} \in \mathrm{AG}(\nu)$, for any budget n of size at least

$$\ddot{c}\theta_{\mathcal{P}_{XY}}(\nu+\varepsilon)\left(\frac{\nu^2}{\varepsilon^2} + \mathrm{Log}\left(\frac{1}{\varepsilon}\right)\right)\left(d\mathrm{Log}\left(\theta_{\mathcal{P}_{XY}}(\nu+\varepsilon)\right) + \mathrm{Log}\left(\frac{\mathrm{Log}(1/\varepsilon)}{\delta}\right)\right), \tag{59}$$

the algorithm produces a classifier $\hat{h}_n$ with $\mathrm{er}_{\mathcal{P}_{XY}}(\hat{h}_n) - \inf_{h\in\mathbb{C}} \mathrm{er}_{\mathcal{P}_{XY}}(h) \leq \varepsilon$ with probability at least $1-\delta$, and requests a number of labels at most n (see also Dasgupta, Hsu, and Monteleoni, 2007; Beygelzimer, Dasgupta, and Langford, 2009, for similar results for related methods). By Theorem 10,

$$\theta_{\mathcal{P}_{XY}}(\nu+\varepsilon) = \theta_{\mathcal{P}_{XY}}((\nu+\varepsilon)\wedge 1) \leq \min\left\{\mathfrak{s}, \frac{1}{(\nu+\varepsilon)\wedge 1}\right\} \leq \min\left\{\mathfrak{s}, \frac{2}{\nu+\varepsilon}\right\} \leq 2\min\left\{\mathfrak{s}, \frac{1}{\nu+\varepsilon}\right\},$$

while $\mathrm{Log}\left(\theta_{\mathcal{P}_{XY}}(\nu+\varepsilon)\right) \leq \mathrm{Log}\left(\min\left\{\mathfrak{s}, \frac{1}{\nu+\varepsilon}\right\} \vee 1\right) = \mathrm{Log}\left(\min\left\{\mathfrak{s}, \frac{1}{\nu+\varepsilon}\right\}\right)$. Therefore, (59) is at most

$$2\ddot{c}\min\left\{\mathfrak{s}, \frac{1}{\nu+\varepsilon}\right\}\left(\frac{\nu^2}{\varepsilon^2} + \mathrm{Log}\left(\frac{1}{\varepsilon}\right)\right)\left(d\mathrm{Log}\left(\min\left\{\mathfrak{s}, \frac{1}{\nu+\varepsilon}\right\}\right) + \mathrm{Log}\left(\frac{\mathrm{Log}(1/\varepsilon)}{\delta}\right)\right),$$

which is therefore an upper bound on $\Lambda_{\mathrm{AG}(\nu)}(\varepsilon,\delta)$. To match the form of the upper bound stated in Theorem 8, we can relax this by noting that $d\mathrm{Log}\left(\min\left\{\mathfrak{s}, \frac{1}{\nu+\varepsilon}\right\}\right) + \mathrm{Log}\left(\frac{\mathrm{Log}(1/\varepsilon)}{\delta}\right) \leq 2d\mathrm{Log}\left(\frac{1}{\varepsilon}\right) + \mathrm{Log}\left(\frac{1}{\delta}\right) \leq 2d\mathrm{Log}\left(\frac{1}{\varepsilon\delta}\right)$, while $\frac{\nu^2}{\varepsilon^2} + \mathrm{Log}\left(\frac{1}{\varepsilon}\right) \leq \left(\frac{\nu^2}{\varepsilon^2}+1\right)\mathrm{Log}\left(\frac{1}{\varepsilon}\right)$.

To prove the lower bound in Theorem 8, we note that $\mathrm{BE}(\nu) \subseteq \mathrm{AG}(\nu)$ for $\nu \in [0, 1/2)$, so that $\Lambda_{\mathrm{BE}(\nu)}(\varepsilon, \delta) \leq \Lambda_{\mathrm{AG}(\nu)}(\varepsilon, \delta)$. Thus, the lower bound on $\Lambda_{\mathrm{BE}(\nu)}(\varepsilon, \delta)$ in Theorem 7 (proven above) also applies to $\Lambda_{\mathrm{AG}(\nu)}(\varepsilon, \delta)$. ■

Appendix C. Proofs for Results in Section 7

This section provides proofs of the equivalences between complexity measures stated in Section 7.

C.1 The Disagreement Coefficient

Here we present the proof of Theorem 10. First, we have a helpful lemma, which allows us to restrict focus to *finitely discrete* probability measures. Let Π denote the set of probability measures $\mathcal{P}$ on $\mathcal{X}$ such that $\exists m \in \mathbb{N}$ and a sequence $\{z_i\}_{i=1}^m$ in $\mathcal{X}$ for which $\mathcal{P}(\{z_i : i \in \{1, \ldots, m\}\}) = 1$.

Lemma 42 *If $\mathfrak{s} < \infty$, then $\forall \varepsilon \in (0, 1]$, $\hat{\hat{\theta}}(\varepsilon) = \sup_{\mathcal{P} \in \Pi} \sup_{h \in \mathbb{C}} \theta_{h,\mathcal{P}}(\varepsilon)$.*

Proof Suppose $\mathfrak{s} < \infty$, and fix any $\varepsilon \in (0, 1]$. Since $\mathcal{P}_{XY}$ ranges over all probability measures over $\mathcal{X} \times \mathcal{Y}$ in the definition of $\hat{\hat{\theta}}(\varepsilon)$, including all those in RE with marginal $\mathcal{P}$ over $\mathcal{X}$ contained in Π (in which case, $\theta_{\mathcal{P}_{XY}}(\varepsilon) = \theta_{f^\star_{\mathcal{P}_{XY}}, \mathcal{P}}(\varepsilon)$), we always have $\sup_{P \in \Pi} \sup_{h \in \mathbb{C}} \theta_{h,P}(\varepsilon) \leq \hat{\hat{\theta}}(\varepsilon)$. Thus, it suffices to show that we also have $\sup_{\mathcal{P} \in \Pi} \sup_{h \in \mathbb{C}} \theta_{h,\mathcal{P}}(\varepsilon) \geq \hat{\hat{\theta}}(\varepsilon)$.

The result trivially holds if $\hat{\hat{\theta}}(\varepsilon) = 1$, since *every* $\mathcal{P}$ and h have $\theta_{h,\mathcal{P}}(\varepsilon) \geq 1$. To address the nontrivial case, suppose $\hat{\hat{\theta}}(\varepsilon) > 1$. Fix any $\gamma_1, \gamma_2, \gamma_3 \in (0, 1)$. Fix any $\mathcal{P}_{XY}$ with $\theta_{\mathcal{P}_{XY}}(\varepsilon) > 1$, and as usual denote $\mathcal{P}(\cdot) = \mathcal{P}_{XY}(\cdot \times \mathcal{Y})$. Also let $h^*_{\mathcal{P}_{XY}}$ be as in Definition 9, so that $\theta_{\mathcal{P}_{XY}}(\varepsilon) = \theta_{h^*_{\mathcal{P}_{XY}}, \mathcal{P}}(\varepsilon)$. Let $r_\varepsilon \in (\varepsilon, 1]$ be such that $\frac{1}{r_\varepsilon} \mathcal{P}(\mathrm{DIS}(\mathrm{B}_{\mathcal{P}}(h^*_{\mathcal{P}_{XY}}, r_\varepsilon))) \geq (1 - \gamma_1)\theta_{\mathcal{P}_{XY}}(\varepsilon)$ (which exists, by the definition of the supremum, combined with the fact that $1 < \theta_{\mathcal{P}_{XY}}(\varepsilon) \leq 1/\varepsilon < \infty$). Also let $h \in \mathbb{C}$ have $\mathcal{P}(x : h(x) \neq h^*_{\mathcal{P}_{XY}}(x)) \leq \gamma_3 r_\varepsilon$, which exists by the definition of $h^*_{\mathcal{P}_{XY}}$.

Let $m = \left\lceil \frac{8}{\gamma_2^2 r_\varepsilon^2} \left(10 d \mathrm{Log}\left(\frac{8e}{\gamma_2^2 r_\varepsilon^2} \right) + \mathrm{Log}(24) \right) \right\rceil$, which is a finite natural number, since $d \leq \mathfrak{s} < \infty$. It follows from Lemma 20 and Lemma 18 that, for $X'_1, \ldots, X'_m$ independent $\mathcal{P}$-distributed random variables, with probability at least $2/3$, every $g \in \mathbb{C}$ has $\frac{1}{m} \sum_{i=1}^m \mathbb{1}_{\mathrm{DIS}(\{h,g\})}(X'_i) \leq \mathcal{P}(x : h(x) \neq g(x)) + \gamma_2 r_\varepsilon \leq \mathcal{P}(x : h^*_{\mathcal{P}_{XY}}(x) \neq g(x)) + (\gamma_3 + \gamma_2) r_\varepsilon$. Furthermore, by Hoeffding's inequality, we also have that with probability at least $2/3$, $\frac{1}{m} \sum_{i=1}^m \mathbb{1}_{\mathrm{DIS}(\mathrm{B}_{\mathcal{P}}(h^*_{\mathcal{P}_{XY}}, r_\varepsilon))}(X'_i) \geq \mathcal{P}(\mathrm{DIS}(\mathrm{B}_{\mathcal{P}}(h^*_{\mathcal{P}_{XY}}, r_\varepsilon))) - \gamma_2 r_\varepsilon$. By a union bound, both of these events happen with probability at least $1/3$. In particular, this implies $\exists z_1, \ldots, z_m \in \mathcal{X}$ such that, letting $\hat{\mathcal{P}}$ be the probability measure with $\hat{\mathcal{P}}(A) = \frac{1}{m} \sum_{i=1}^m \mathbb{1}_A(z_m)$ for all measurable $A \subseteq \mathcal{X}$, we have, $\forall g \in \mathbb{C}$, $\hat{\mathcal{P}}(\mathrm{DIS}(\{h, g\})) \leq \mathcal{P}(\mathrm{DIS}(\{h^*_{\mathcal{P}_{XY}}, g\})) + (\gamma_3 + \gamma_2) r_\varepsilon$, and furthermore $\hat{\mathcal{P}}(\mathrm{DIS}(\mathrm{B}_{\mathcal{P}}(h^*_{\mathcal{P}_{XY}}, r_\varepsilon))) \geq \mathcal{P}(\mathrm{DIS}(\mathrm{B}_{\mathcal{P}}(h^*_{\mathcal{P}_{XY}}, r_\varepsilon))) - \gamma_2 r_\varepsilon$. This further implies

that $\mathrm{B}_{\mathcal{P}}(h^*_{\mathcal{P}_{XY}}, r_\varepsilon) \subseteq \mathrm{B}_{\hat{\mathcal{P}}}(h, (1+\gamma_3+\gamma_2)r_\varepsilon)$, and thus

$$\begin{aligned}\hat{\mathcal{P}}(\mathrm{DIS}(\mathrm{B}_{\hat{\mathcal{P}}}(h,(1+\gamma_3+\gamma_2)r_\varepsilon))) &\geq \hat{\mathcal{P}}(\mathrm{DIS}(\mathrm{B}_{\mathcal{P}}(h^*_{\mathcal{P}_{XY}}, r_\varepsilon))) \geq \mathcal{P}(\mathrm{DIS}(\mathrm{B}_{\mathcal{P}}(h^*_{\mathcal{P}_{XY}}, r_\varepsilon))) - \gamma_2 r_\varepsilon \\ &\geq (1-\gamma_1)\theta_{\mathcal{P}_{XY}}(\varepsilon) r_\varepsilon - \gamma_2 r_\varepsilon \geq (1-\gamma_1-\gamma_2)\theta_{\mathcal{P}_{XY}}(\varepsilon) r_\varepsilon.\end{aligned}$$

Therefore,

$$\theta_{h,\hat{\mathcal{P}}}(\varepsilon) \geq \frac{\hat{\mathcal{P}}(\mathrm{DIS}(\mathrm{B}_{\hat{\mathcal{P}}}(h,(1+\gamma_3+\gamma_2)r_\varepsilon)))}{(1+\gamma_3+\gamma_2)r_\varepsilon} \geq \frac{1-\gamma_1-\gamma_2}{1+\gamma_3+\gamma_2}\theta_{\mathcal{P}_{XY}}(\varepsilon).$$

Noting that $\hat{\mathcal{P}}(\{z_1,\ldots,z_m\}) = 1$, so that $\hat{\mathcal{P}} \in \Pi$, since $\mathcal{P}_{XY}$ was arbitrary, we have established that $\forall \mathcal{P}_{XY}$, $\exists P \in \Pi$ and $h \in \mathbb{C}$ such that $\theta_{h,P}(\varepsilon) \geq \frac{1-\gamma_1-\gamma_2}{1+\gamma_3+\gamma_2}\theta_{\mathcal{P}_{XY}}(\varepsilon)$. Since this holds for any choices of $\gamma_1, \gamma_2, \gamma_3 \in (0,1)$, taking the limits as $\gamma_1 \to 0$, $\gamma_3 \to 0$, and $\gamma_2 \to 0$, we have $\sup_{P\in\Pi}\sup_{h\in\mathbb{C}}\theta_{h,P}(\varepsilon) \geq \hat{\hat{\theta}}(\varepsilon)$. ■

In fact, it is easy to show (based on the first part of the proof below) that the "$\mathfrak{s} < \infty$" constraint is unnecessary in Lemma 42, though this is not important for our purposes. We are now ready for the proof of Theorem 10.

Proof of Theorem 10 First, we prove $\hat{\hat{\theta}}(\varepsilon) \geq \mathfrak{s} \wedge \frac{1}{\varepsilon}$. Toward this end, let $\{x_i\}_{i=1}^{\mathfrak{s}}$ and $\{h_i\}_{i=0}^{\mathfrak{s}}$ be as in Definition 2, and let $m = \mathfrak{s} \wedge \lceil \frac{1}{\varepsilon} \rceil$. Let $\mathcal{P}$ be a probability measure on $\mathcal{X}$ with $\mathcal{P}(\{x_i\}) = 1/m$ for each $i \in \{1,\ldots,m\}$. In particular, this implies that every $i \in \{1,\ldots,m\}$ has $\mathcal{P}(x : h_i(x) \neq h_0(x)) = 1/m$, so that $h_i \in \mathrm{B}_{\mathcal{P}}(h_0, 1/m)$. Since clearly $h_0 \in \mathrm{B}_{\mathcal{P}}(h_0, 1/m)$ as well, and every $i \in \{1,\ldots,m\}$ has $x_i \in \mathrm{DIS}(\{h_i, h_0\})$, every $r > 1/m$ has $\mathcal{P}(\mathrm{DIS}(\mathrm{B}_{\mathcal{P}}(h_0, r))) = \mathcal{P}(\{x_i : i \in \{1,\ldots,m\}\}) = 1$. Therefore, letting $\mathcal{P}_{XY}$ be the distribution in RE with $f^\star_{\mathcal{P}_{XY}} = h_0$ and marginal $\mathcal{P}$ over $\mathcal{X}$,

$$\begin{aligned}\hat{\hat{\theta}}(\varepsilon) \geq \theta_{\mathcal{P}_{XY}}(\varepsilon) = \theta_{h_0,\mathcal{P}}(\varepsilon) &\geq \frac{\mathcal{P}(\mathrm{DIS}(\mathrm{B}_{\mathcal{P}}(h_0, \max\{1/m, \varepsilon\})))}{\max\{1/m,\varepsilon\}} \\ &= \frac{1}{\max\{1/m,\varepsilon\}} = m \wedge \frac{1}{\varepsilon} = \mathfrak{s} \wedge \frac{1}{\varepsilon}.\end{aligned}$$

Next, we prove that $\hat{\hat{\theta}}(\varepsilon) \leq \mathfrak{s} \wedge \frac{1}{\varepsilon}$. That $\hat{\hat{\theta}}(\varepsilon) \leq \frac{1}{\varepsilon}$ follows directly from the definition, and the fact that probabilities are at most 1: that is, any $\mathcal{P}$ and h have $\sup_{r>\varepsilon}\frac{\mathcal{P}(\mathrm{DIS}(\mathrm{B}_{\mathcal{P}}(h,r)))}{r} \leq \sup_{r>\varepsilon}\frac{1}{r} = \frac{1}{\varepsilon}$. Therefore, it remains only to show that $\hat{\hat{\theta}}(\varepsilon) \leq \mathfrak{s}$ when $\mathfrak{s} < \frac{1}{\varepsilon}$. Furthermore, Lemma 42 implies that it suffices to show that $\sup_{\mathcal{P}\in\Pi}\sup_{h\in\mathbb{C}}\theta_{h,\mathcal{P}}(\varepsilon) \leq \mathfrak{s}$ in this case. Toward this end, suppose $\mathfrak{s} < \frac{1}{\varepsilon}$. We first stratify the set Π based on the size of the support, defining, for each $m \in \mathbb{N}$, $\Pi_m = \{\mathcal{P} \in \Pi : \exists z_1,\ldots,z_m \in \mathcal{X} \text{ s.t. } \mathcal{P}(\{z_1,\ldots,z_m\}) = 1\}$. Thus, Π_m is the set of probability measures on $\mathcal{X}$ for which the support of the probability mass function has cardinality at most m.

We now proceed by induction on m. As a base case, fix any $m \leq \mathfrak{s}$, any classifier h, and any $\mathcal{P} \in \Pi_m$, and let $z_1,\ldots,z_m \in \mathcal{X}$ be such that $\mathcal{P}(\{z_1,\ldots,z_m\}) = 1$. For any $r \in [1/\mathfrak{s}, 1]$, $\mathcal{P}(\mathrm{DIS}(\mathrm{B}_{\mathcal{P}}(h,r)))/r \leq 1/r \leq \mathfrak{s}$. Furthermore (following an argument of Hanneke, 2014), for any $r \in (\varepsilon, 1/\mathfrak{s})$, for any $g \in \mathbb{C}$ with $\mathcal{P}(x : g(x) \neq h(x)) \leq r$, every $z \in \mathcal{X}$ with $\mathcal{P}(\{z\}) > r$ has $\mathcal{P}(x : g(x) \neq h(x)) < \mathcal{P}(\{z\})$, so that $g(z) = h(z)$; thus, $z \notin \mathrm{DIS}(\mathrm{B}_{\mathcal{P}}(h,r))$. We therefore have that $\mathcal{P}(\mathrm{DIS}(\mathrm{B}_{\mathcal{P}}(h,r))) \leq \mathcal{P}(x : \mathcal{P}(\{x\}) \leq r) = \sum_{i=1}^m \mathbb{1}[\mathcal{P}(\{z_i\}) \leq r]\mathcal{P}(\{z_i\}) \leq$

$r|\{i \in \{1,\dots,m\} : \mathcal{P}(\{z_i\}) \le r\}|$. Therefore, $\frac{\mathcal{P}(\mathrm{DIS}(\mathrm{B}_{\mathcal{P}}(h,r)))}{r} \le |\{i \in \{1,\dots,m\} : \mathcal{P}(\{z_i\}) \le r\}| \le m \le \mathfrak{s}$, so that (since $\mathfrak{s} \ge 1$, due to the assumption that $|\mathbb{C}| \ge 2$), we have $\theta_{h,\mathcal{P}}(\varepsilon) \le \mathfrak{s}$.

Now take as an inductive hypothesis that, for some $m \in \mathbb{N}$ with $m > \mathfrak{s}$, we have

$$\sup_{\mathcal{P} \in \Pi_{m-1}} \sup_{h \in \mathbb{C}} \theta_{h,\mathcal{P}}(\varepsilon) \le \mathfrak{s}.$$

Fix any $h \in \mathbb{C}$, $r > \varepsilon$, and $\mathcal{P} \in \Pi_m$, and let $z_1,\dots,z_m \in \mathcal{X}$ be such that $\mathcal{P}(\{z_1,\dots,z_m\}) = 1$. If $\exists i,j \in \{1,\dots,m\}$ with $i \ne j$ and $z_i = z_j$, or if some $j \in \{1,\dots,m\}$ has $\mathcal{P}(\{z_j\}) = 0$, then since either of these has $\mathcal{P}(\{z_k : k \in \{1,\dots,m\} \setminus \{j\}\}) = 1$, we would also have $\mathcal{P} \in \Pi_{m-1}$, so that $\theta_{h,\mathcal{P}}(\varepsilon) \le \mathfrak{s}$ by the inductive hypothesis. To handle the remaining nontrivial cases, suppose the $z_1,\dots,z_m$ are all distinct, and $\min_{i \in \{1,\dots,m\}} \mathcal{P}(\{z_i\}) > 0$. Furthermore, note that, since $m > \mathfrak{s}$, $\{z_1,\dots,z_m\}$ cannot be a star set for $\mathbb{C}$.

We now consider three cases. First, consider the case that $\exists k \in \{1,\dots,m\}$ with $z_k \notin \mathrm{DIS}(\mathrm{B}_{\mathcal{P}}(h,r))$. In this case, define a probability measure $\mathcal{P}'$ over $\mathcal{X}$ such that, for any measurable $A \subseteq \mathcal{X} \setminus \{z_k\}$, $\mathcal{P}'(A) = \mathcal{P}'(A \cup \{z_k\}) = \mathcal{P}(A)/(1-\mathcal{P}(\{z_k\}))$. Note that this is a well-defined probability measure, since $m \ge 2$ and $\min_{i \in \{1,\dots,m\}} \mathcal{P}(\{z_i\}) > 0$, so that $\mathcal{P}(\mathcal{X} \setminus \{z_k\}) = 1 - \mathcal{P}(\{z_k\}) > 0$. Also note that (since $h \in \mathrm{B}_{\mathcal{P}}(h,r)$) any $g \in \mathrm{B}_{\mathcal{P}}(h,r)$ has $g(z_k) = h(z_k)$, so that $\mathcal{P}'(x : g(x) \ne h(x)) = \mathcal{P}(x : g(x) \ne h(x))/(1-\mathcal{P}(\{z_k\})) \le r/(1-\mathcal{P}(\{z_k\}))$. Therefore, $\mathrm{B}_{\mathcal{P}'}(h, r/(1-\mathcal{P}(\{z_k\}))) \supseteq \mathrm{B}_{\mathcal{P}}(h,r)$, and since $z_k \notin \mathrm{DIS}(\mathrm{B}_{\mathcal{P}}(h,r))$, $\mathcal{P}'(\mathrm{DIS}(\mathrm{B}_{\mathcal{P}'}(h, r/(1-\mathcal{P}(\{z_k\}))))) \ge \mathcal{P}'(\mathrm{DIS}(\mathrm{B}_{\mathcal{P}}(h,r))) = \mathcal{P}(\mathrm{DIS}(\mathrm{B}_{\mathcal{P}}(h,r)))/(1-\mathcal{P}(\{z_k\}))$. Thus,

$$\mathcal{P}(\mathrm{DIS}(\mathrm{B}_{\mathcal{P}}(h,r))) \le (1-\mathcal{P}(\{z_k\}))\mathcal{P}'(\mathrm{DIS}(\mathrm{B}_{\mathcal{P}'}(h, r/(1-\mathcal{P}(\{z_k\})))))). \tag{60}$$

Noting that $\mathcal{P}'(\{z_i : i \in \{1,\dots,m\} \setminus \{k\}\}) = \mathcal{P}(\{z_1,\dots,z_m\} \setminus \{z_k\})/(1-\mathcal{P}(\{z_k\})) = 1$, we have that $\mathcal{P}' \in \Pi_{m-1}$. Therefore, by the inductive hypothesis and the fact that $r/(1-\mathcal{P}(\{z_k\})) > r > \varepsilon$,

$$\begin{aligned}\mathcal{P}'\left(\mathrm{DIS}\left(\mathrm{B}_{\mathcal{P}'}\left(h, \frac{r}{1-\mathcal{P}(\{z_k\})}\right)\right)\right) &\le \theta_{h,\mathcal{P}'}(\varepsilon)\frac{r}{1-\mathcal{P}(\{z_k\})} \\ &\le \sup_{P \in \Pi_{m-1}} \sup_{h' \in \mathbb{C}} \theta_{h',P}(\varepsilon)\frac{r}{1-\mathcal{P}(\{z_k\})} \le \frac{\mathfrak{s}r}{1-\mathcal{P}(\{z_k\})}.\end{aligned}$$

Combined with (60), this further implies that $\mathcal{P}(\mathrm{DIS}(\mathrm{B}_{\mathcal{P}}(h,r))) \le (1-\mathcal{P}(\{z_k\}))\mathfrak{s}r/(1-\mathcal{P}(\{z_k\})) = \mathfrak{s}r$.

Next, consider a second case, where $\{z_1,\dots,z_m\} \subseteq \mathrm{DIS}(\mathrm{B}_{\mathcal{P}}(h,r))$, and $\exists j,k \in \{1,\dots,m\}$ with $j \ne k$ such that, $\forall g \in \mathrm{B}_{\mathcal{P}}(h,r)$, $g(z_k) \ne h(z_k) \Rightarrow g(z_j) \ne h(z_j)$. In this case, define a probability measure $\mathcal{P}'$ over $\mathcal{X}$ such that, for any measurable $A \subseteq \mathcal{X} \setminus \{z_j, z_k\}$, $\mathcal{P}'(A) = \mathcal{P}(A)$, $\mathcal{P}'(A \cup \{z_j\}) = \mathcal{P}(A)$, and $\mathcal{P}'(A \cup \{z_k\}) = \mathcal{P}'(A \cup \{z_j, z_k\}) = \mathcal{P}(A \cup \{z_j, z_k\})$: in other words, $\mathcal{P}'$ has a probability mass function $x \mapsto \mathcal{P}'(\{x\})$ equal to $x \mapsto \mathcal{P}(\{x\})$ everywhere, except that $\mathcal{P}'(\{z_j\}) = 0$ and $\mathcal{P}'(\{z_k\}) = \mathcal{P}(\{z_j\}) + \mathcal{P}(\{z_k\})$. Note that, for any $g \in \mathrm{B}_{\mathcal{P}}(h,r)$ with $g(z_k) = h(z_k)$, $\mathcal{P}'(x : g(x) \ne h(x)) = \mathcal{P}(x : g(x) \ne h(x)) - \mathbb{1}[g(z_j) \ne h(z_j)]\mathcal{P}(\{z_j\}) \le \mathcal{P}(x : g(x) \ne h(x)) \le r$. Furthermore, any $g \in \mathrm{B}_{\mathcal{P}}(h,r)$ with $g(z_k) \ne h(z_k)$ also has $g(z_j) \ne h(z_j)$, so that $\mathcal{P}'(x : g(x) \ne h(x)) = \mathcal{P}(x : g(x) \ne h(x)) \le r$. Therefore, $\mathrm{B}_{\mathcal{P}'}(h,r) \supseteq \mathrm{B}_{\mathcal{P}}(h,r)$. Since $z_j, z_k \in \mathrm{DIS}(\mathrm{B}_{\mathcal{P}}(h,r))$, this further implies that $z_j, z_k \in \mathrm{DIS}(\mathrm{B}_{\mathcal{P}'}(h,r))$. Therefore, by definition of $\mathcal{P}'$ and monotonicity of measures, $\mathcal{P}'(\mathrm{DIS}(\mathrm{B}_{\mathcal{P}'}(h,r))) = \mathcal{P}(\mathrm{DIS}(\mathrm{B}_{\mathcal{P}'}(h,r))) \ge \mathcal{P}(\mathrm{DIS}(\mathrm{B}_{\mathcal{P}}(h,r)))$. Noting that $\mathcal{P}'(\{z_i : i \in$

$\{1,\dots,m\} \setminus \{j\}) = \mathcal{P}(\{z_1,\dots,z_m\}) = 1$, we have $\mathcal{P}' \in \Pi_{m-1}$, and therefore (by the inductive hypothesis), $\mathcal{P}'(\mathrm{DIS}(\mathrm{B}_{\mathcal{P}'}(h,r))) \leq \theta_{h,\mathcal{P}'}(\varepsilon)r \leq \sup_{P\in\Pi_{m-1}} \sup_{h'\in\mathbb{C}} \theta_{h',P}(\varepsilon)r \leq \mathfrak{s}r$. Thus, since we established above that $\mathcal{P}(\mathrm{DIS}(\mathrm{B}_{\mathcal{P}}(h,r))) \leq \mathcal{P}'(\mathrm{DIS}(\mathrm{B}_{\mathcal{P}'}(h,r)))$, we have that $\mathcal{P}(\mathrm{DIS}(\mathrm{B}_{\mathcal{P}}(h,r))) \leq \mathfrak{s}r$.

Finally, consider a third case (the complement of the first two), in which $\{z_1,\dots,z_m\} \subseteq \mathrm{DIS}(\mathrm{B}_{\mathcal{P}}(h,r))$, but $\nexists j,k \in \{1,\dots,m\}$ with $j \neq k$ such that, $\forall g \in \mathrm{B}_{\mathcal{P}}(h,r)$, $g(z_k) \neq h(z_k) \Rightarrow g(z_j) \neq h(z_j)$. In particular, note that the first condition (which is, in fact, redundant, but included for clarity) implies $\mathcal{P}(\mathrm{DIS}(\mathrm{B}_{\mathcal{P}}(h,r))) = 1$. In this case, since (as above) $\{z_1,\dots,z_m\}$ is not a star set for $\mathbb{C}$, $\exists i \in \{1,\dots,m\}$ such that $\forall g \in \mathbb{C}$ with $g(z_i) \neq h(z_i)$, $\exists j \in \{1,\dots,m\} \setminus \{i\}$ with $g(z_j) \neq h(z_j)$ as well; fix any such $i \in \{1,\dots,m\}$. Since $\{z_1,\dots,z_m\} \subseteq \mathrm{DIS}(\mathrm{B}_{\mathcal{P}}(h,r))$, we have $z_i \in \mathrm{DIS}(\mathrm{B}_{\mathcal{P}}(h,r))$. Thus, we may let $g_i \in \mathrm{B}_{\mathcal{P}}(h,r)$ be such that $g_i(z_i) \neq h(z_i)$, and let $j \in \{1,\dots,m\} \setminus \{i\}$ be such that $g_i(z_j) \neq h(z_j)$ (which exists, by our choice of i). Let $\mathcal{P}'$ be a probability measure over $\mathcal{X}$ such that, for all measurable $A \subseteq \mathcal{X} \setminus \{z_i,z_j\}$, $\mathcal{P}'(A) = \mathcal{P}(A)$, $\mathcal{P}'(A \cup \{z_i\}) = \mathcal{P}(A)$, and $\mathcal{P}'(A \cup \{z_j\}) = \mathcal{P}'(A \cup \{z_i,z_j\}) = \mathcal{P}(A \cup \{z_i,z_j\})$: in other words, $\mathcal{P}'$ has a probability mass function $x \mapsto \mathcal{P}'(\{x\})$ equal to $x \mapsto \mathcal{P}(\{x\})$ everywhere, except that $\mathcal{P}'(\{z_i\}) = 0$ and $\mathcal{P}'(\{z_j\}) = \mathcal{P}(\{z_i\}) + \mathcal{P}(\{z_j\})$. Note that, for any measurable set $A \subseteq \mathcal{X}$ with $\{z_i,z_j\} \subseteq A$, $\mathcal{P}'(A) = \mathcal{P}(A)$. In particular, since $\{z_i,z_j\} \subseteq \mathrm{DIS}(\{g_i,h\})$, $\mathcal{P}'(\mathrm{DIS}(\{g_i,h\})) = \mathcal{P}(\mathrm{DIS}(\{g_i,h\})) \leq r$, so that $g_i \in \mathrm{B}_{\mathcal{P}'}(h,r)$, and therefore (since $h \in \mathrm{B}_{\mathcal{P}'}(h,r)$ as well) $\{z_i,z_j\} \subseteq \mathrm{DIS}(\mathrm{B}_{\mathcal{P}'}(h,r))$. Furthermore, for any $k \in \{1,\dots,m\} \setminus \{i,j\}$, by the property characterizing this third case, and since $z_k \in \mathrm{DIS}(\mathrm{B}_{\mathcal{P}}(h,r))$, $\exists g \in \mathrm{B}_{\mathcal{P}}(h,r)$ with $g(z_k) \neq h(z_k)$ and $g(z_j) = h(z_j)$, so that $\mathcal{P}'(\mathrm{DIS}(\{g,h\})) = \mathcal{P}(\mathrm{DIS}(\{g,h\}) \setminus \{z_i\}) \leq \mathcal{P}(\mathrm{DIS}(\{g,h\})) \leq r$ (i.e., $g \in \mathrm{B}_{\mathcal{P}'}(h,r)$), and therefore (since $h \in \mathrm{B}_{\mathcal{P}'}(h,r)$ as well) $z_k \in \mathrm{DIS}(\mathrm{B}_{\mathcal{P}'}(h,r))$ as well. Altogether, we have that $\{z_1,\dots,z_m\} \subseteq \mathrm{DIS}(\mathrm{B}_{\mathcal{P}'}(h,r))$. Therefore, since $\{z_i,z_j\} \subseteq \mathrm{DIS}(\mathrm{B}_{\mathcal{P}'}(h,r))$, the definition of $\mathcal{P}'$ implies $\mathcal{P}'(\mathrm{DIS}(\mathrm{B}_{\mathcal{P}'}(h,r))) = \mathcal{P}(\mathrm{DIS}(\mathrm{B}_{\mathcal{P}'}(h,r))) \geq \mathcal{P}(\{z_1,\dots,z_m\}) = 1 = \mathcal{P}(\mathrm{DIS}(\mathrm{B}_{\mathcal{P}}(h,r)))$. Noting that $\mathcal{P}'(\{z_k : k \in \{1,\dots,m\} \setminus \{i\}\}) = \mathcal{P}(\{z_1,\dots,z_m\}) = 1$, we have that $\mathcal{P}' \in \Pi_{m-1}$, and therefore (by the inductive hypothesis), $\mathcal{P}'(\mathrm{DIS}(\mathrm{B}_{\mathcal{P}'}(h,r))) \leq \theta_{h,\mathcal{P}'}(\varepsilon)r \leq \sup_{P\in\Pi_{m-1}} \sup_{h'\in\mathbb{C}} \theta_{h',P}(\varepsilon)r \leq \mathfrak{s}r$. Since $\mathcal{P}'(\mathrm{DIS}(\mathrm{B}_{\mathcal{P}'}(h,r))) = 1 = \mathcal{P}(\mathrm{DIS}(\mathrm{B}_{\mathcal{P}}(h,r)))$, we have that $\mathcal{P}(\mathrm{DIS}(\mathrm{B}_{\mathcal{P}}(h,r))) \leq \mathfrak{s}r$ as well.

Thus, in all three cases, we have that $\mathcal{P}(\mathrm{DIS}(\mathrm{B}_{\mathcal{P}}(h,r))) \leq \mathfrak{s}r$. Since this holds for every $r > \varepsilon$, and $|\mathbb{C}| \geq 2$ implies $\mathfrak{s} \geq 1$, we have that $\theta_{h,\mathcal{P}}(\varepsilon) \leq \mathfrak{s}$. Since this holds for every $h \in \mathbb{C}$ and $\mathcal{P} \in \Pi_m$, we have established that $\sup_{\mathcal{P}\in\Pi_m} \sup_{h\in\mathbb{C}} \theta_{h,\mathcal{P}}(\varepsilon) \leq \mathfrak{s}$, which completes the inductive step. It follows by the principle of induction that $\sup_{\mathcal{P}\in\Pi_m} \sup_{h\in\mathbb{C}} \theta_{h,\mathcal{P}}(\varepsilon) \leq \mathfrak{s}$ for every $m \in \mathbb{N}$, and therefore, since $\Pi = \bigcup_m \Pi_m$, $\sup_{\mathcal{P}\in\Pi} \sup_{h\in\mathbb{C}} \theta_{h,\mathcal{P}}(\varepsilon) \leq \mathfrak{s}$.

The claim that $\hat{\hat{\theta}}(0) = \mathfrak{s}$ follows as a limiting case, due to continuity of the supremum from below. Specifically, fix any sequence $\{A_n\}_{n=1}^{\infty}$ of nonempty subsets of $\mathbb{R}$. For each $m \in \mathbb{N}$, $\bigcup_n A_n \supseteq A_m$, so $\sup \bigcup_n A_n \geq \sup A_m$ (allowing the supremum to take the value ∞ where appropriate), and since this holds for every such m, we have $\sup \bigcup_n A_n \geq \sup_n \sup A_n$ Furthermore, $\forall a \in \bigcup_n A_n$, $\exists m \in \mathbb{N}$ s.t. $a \in A_m$, so that $\sup_n \sup A_n \geq \sup A_m \geq a$, and therefore (since this holds for every such a) $\sup_n \sup A_n \geq \sup \bigcup_n A_n$. Thus, $\sup \bigcup_n A_n = \sup_n \sup A_n$. In particular, taking (for each $n \in \mathbb{N}$)

$$A_n = \left\{ \frac{\mathcal{P}(\mathrm{DIS}(\mathrm{B}_{\mathcal{P}}(h^*_{\mathcal{P}_{XY}},r)))}{r} \vee 1 : r > 1/n, \mathcal{P}_{XY} \in \mathrm{AG}(1) \right\},$$

(where, as usual, $\mathcal{P}(\cdot) = \mathcal{P}_{XY}(\cdot \times \mathcal{Y})$ denotes the marginal of $\mathcal{P}_{XY}$ over $\mathcal{X}$), and noting that $\sup \bigcup_n A_n = \hat{\hat{\theta}}(0)$ and $\forall n \in \mathbb{N}$, $\sup A_n = \hat{\hat{\theta}}(1/n)$, we have that $\hat{\hat{\theta}}(0) = \sup_n \hat{\hat{\theta}}(1/n) = \sup_n \mathfrak{s} \wedge n = \mathfrak{s}$. ■

C.2 The Splitting Index

Here we present the proof of Theorem 12. First, we introduce a quantity related to $\hat{\bar{\rho}}(\varepsilon)$, but slightly simpler. For $\varepsilon, \tau \in (0,1]$ and any probability measure $\mathcal{P}$ over $\mathcal{X}$, define

$$\bar{\rho}_{\mathcal{P}}(\varepsilon;\tau) = \sup\left\{\rho \in [0,1] : \mathbb{C} \text{ is } (\rho,\varepsilon,\tau)\text{-splittable under } \mathcal{P}\right\},$$

and let

$$\bar{\rho}(\varepsilon) = \inf_P \lim_{\tau \to 0} \bar{\rho}_P(\varepsilon;\tau).$$

In the arguments below, we will see that $\lfloor 1/\bar{\rho}(\varepsilon) \rfloor = \left\lfloor 1/\hat{\bar{\rho}}(\varepsilon) \right\rfloor$, so that it suffices to work with this simpler quantity. We begin with a lemma which allows us to restrict our focus (in part of the proof) to finitely discrete probability measures. Recall the definition of Π from Appendix C.1 above.

Lemma 43 *If $d < \infty$, then $\forall \varepsilon \in (0,1]$, $\bar{\rho}(\varepsilon) \geq \lim_{\gamma \to 0} \inf_{P \in \Pi} \lim_{\tau \to 0} \bar{\rho}_P((1-\gamma)\varepsilon;\tau)$.*

Proof Suppose $d < \infty$, and fix any $\varepsilon \in (0,1]$. Fix arbitrary values $\gamma_1, \gamma_2 \in (0,1)$, and let

$$m = \left\lceil \frac{8}{\gamma_2^2 \varepsilon^2} \left(10d\mathrm{Log}\left(\frac{8e}{\gamma_2^2\varepsilon^2}\right) + \mathrm{Log}(24) \right) \right\rceil,$$

which is a finite natural number. Fix any probability measure $\mathcal{P}$ over $\mathcal{X}$, and any $\tau \in (0, 1/(3m))$, and note that $\tau' \mapsto \bar{\rho}_{\mathcal{P}}(\varepsilon;\tau')$ is nonincreasing, so that $\bar{\rho}_{\mathcal{P}}(\varepsilon;\tau) \leq \lim_{\tau' \to 0} \bar{\rho}_{\mathcal{P}}(\varepsilon;\tau')$. For brevity, denote $\bar{\rho} = \bar{\rho}_{\mathcal{P}}(\varepsilon;\tau)$. Since $\mathbb{C}$ is not $(\gamma_1 + \bar{\rho}, \varepsilon, \tau)$-splittable under $\mathcal{P}$, let $Q \subseteq \{\{f,g\} \subseteq \mathbb{C} : \mathcal{P}(x : f(x) \neq g(x)) \geq \varepsilon\}$ be a finite set such that

$$\mathcal{P}\left(x : \mathrm{Split}(Q,x) \geq (\gamma_1 + \bar{\rho})\,|Q|\right) < \tau.$$

Let $X_1', \ldots, X_m'$ be independent $\mathcal{P}$-distributed random variables. Lemmas 18 and 20 imply that, with probability at least $2/3$, $\forall f, g \in \mathbb{C}$,

$$\left| \mathcal{P}(x : f(x) \neq g(x)) - \frac{1}{m}\sum_{i=1}^m \mathbb{1}\left[f(X_i') \neq g(X_i')\right] \right| \leq \gamma_2 \varepsilon.$$

Furthermore, by a union bound, with probability at least $1 - m\mathcal{P}(x : \mathrm{Split}(Q,x) \geq (\gamma_1+\bar{\rho})\,|Q|) > 1 - m\tau > 1 - m(1/(3m)) = 2/3$, every $i \in \{1,\ldots,m\}$ has $\mathrm{Split}(Q, X_i') < (\gamma_1 + \bar{\rho})|Q|$. By a union bound, both of the above events occur with probability at least $1/3$. In particular, this implies $\exists z_1, \ldots, z_m \in \mathcal{X}$ such that, letting $\hat{\mathcal{P}}$ denote the probability measure with $\hat{\mathcal{P}}(A) = \frac{1}{m}\sum_{i=1}^m \mathbb{1}_A(z_m)$ for all measurable $A \subseteq \mathcal{X}$, we have, $\forall f, g \in \mathbb{C}$, $\left|\mathcal{P}(x : f(x) \neq g(x)) - \hat{\mathcal{P}}(x : f(x) \neq g(x))\right| \leq \gamma_2\varepsilon$, and $\hat{\mathcal{P}}(x : \mathrm{Split}(Q,x) \geq (\gamma_1 + \bar{\rho})|Q|) = 0$.

For any $\{f,g\} \in Q$, we have $\hat{\mathcal{P}}(x : f(x) \neq g(x)) \geq \mathcal{P}(x : f(x) \neq g(x)) - \gamma_2\varepsilon \geq (1-\gamma_2)\varepsilon$. Therefore, $\mathbb{C}$ is not $(\gamma_1 + \bar{\rho}, (1-\gamma_2)\varepsilon, \tau')$-splittable under $\hat{\mathcal{P}}$ for any $\tau' > 0$, which implies $\lim_{\tau'\to 0} \bar{\rho}_{\hat{\mathcal{P}}}((1-\gamma_2)\varepsilon;\tau') \leq \gamma_1 + \bar{\rho}_{\mathcal{P}}(\varepsilon;\tau)$. Since $\hat{\mathcal{P}} \in \Pi$, we have

$$\inf_{P\in\Pi} \lim_{\tau'\to 0} \bar{\rho}_P((1-\gamma_2)\varepsilon;\tau') \leq \gamma_1 + \bar{\rho}_{\mathcal{P}}(\varepsilon;\tau) \leq \gamma_1 + \lim_{\tau'\to 0} \bar{\rho}_{\mathcal{P}}(\varepsilon;\tau').$$

Since this holds for any $\gamma_1 \in (0,1)$, taking the limit as $\gamma_1 \to 0$ implies

$$\inf_{P\in\Pi} \lim_{\tau'\to 0} \bar{\rho}_P((1-\gamma_2)\varepsilon;\tau') \leq \lim_{\tau'\to 0} \bar{\rho}_{\mathcal{P}}(\varepsilon;\tau').$$

Furthermore, since this holds for any $\gamma_2 \in (0,1)$ and any $\mathcal{P}$, we have

$$\lim_{\gamma_2\to 0} \inf_{P\in\Pi} \lim_{\tau'\to 0} \bar{\rho}_P((1-\gamma_2)\varepsilon;\tau') \leq \inf_P \lim_{\tau'\to 0} \bar{\rho}_P(\varepsilon;\tau') = \bar{\rho}(\varepsilon).$$

■

We are now ready for the proof of Theorem 12.

Proof of Theorem 12 We first establish that $\mathfrak{s} \wedge \left\lfloor \frac{1}{\varepsilon} \right\rfloor \leq \left\lfloor \frac{1}{\hat{\bar{\rho}}(\varepsilon)} \right\rfloor$ for any $\varepsilon \in (0,1]$. The proof of this fact was implicitly established in the original work of Dasgupta (2005, Corollary 3), but we include the argument here for completeness. Let $\{x_i\}_{i=1}^{\mathfrak{s}}$ and $\{h_i\}_{i=0}^{\mathfrak{s}}$ be as in Definition 2, and let $m = \mathfrak{s} \wedge \left\lfloor \frac{1}{\varepsilon} \right\rfloor$. Let $\Delta = 1/m$, and note that $\Delta \geq 1/\left\lfloor \frac{1}{\varepsilon} \right\rfloor \geq \varepsilon$. As in the proof of Theorem 10, let $\mathcal{P}$ be a probability measure on $\mathcal{X}$ with $\mathcal{P}(\{x_i\}) = 1/m$ for each $i \in \{1,\ldots,m\}$. Thus, every $i \in \{1,\ldots,m\}$ has $\mathcal{P}(x : h_i(x) \neq h_0(x)) = \Delta$, so that $h_i \in \mathrm{B}_{\mathcal{P}}(h_0,\Delta) \subseteq \mathrm{B}_{\mathcal{P}}(h_0,4\Delta)$, and the finite set $Q = \{\{h_0,h_i\} : i \in \{1,\ldots,m\}\}$ satisfies $Q \subseteq \{\{f,g\} \subseteq \mathrm{B}_{\mathcal{P}}(h_0,4\Delta) : \mathcal{P}(x : f(x) \neq g(x)) \geq \Delta\}$. In particular, since $\mathcal{P}(\mathcal{X} \setminus \{x_1,\ldots,x_m\}) = 0$, and every $i \in \{1,\ldots,m\}$ has $\mathrm{Split}(Q,x_i) = 1 = \frac{1}{m}|Q|$, we have $\mathcal{P}\left(x : \mathrm{Split}(Q,x) > \frac{1}{m}|Q|\right) = 0$. Thus, for any $\rho > \frac{1}{m}$, and any $\tau > 0$, $\mathrm{B}_{\mathcal{P}}(h_0,4\Delta)$ is not (ρ,Δ,τ)-splittable. Therefore, $\hat{\bar{\rho}}(\varepsilon) \leq \lim_{\tau\to 0} \rho_{h_0,\mathcal{P}}(\varepsilon;\tau) \leq \frac{1}{m}$, which implies $\frac{1}{\hat{\bar{\rho}}(\varepsilon)} \geq m$; since $m \in \mathbb{N}$, it follows that $\left\lfloor \frac{1}{\hat{\bar{\rho}}(\varepsilon)} \right\rfloor \geq m$.

Next, we prove that $\left\lfloor \frac{1}{\hat{\bar{\rho}}(\varepsilon)} \right\rfloor \leq \mathfrak{s} \wedge \left\lfloor \frac{1}{\varepsilon} \right\rfloor$ for any $\varepsilon \in (0,1]$. Since, for every $h \in \mathbb{C}$, every probability measure $\mathcal{P}$ over $\mathcal{X}$, and every $\Delta \geq \varepsilon$, every finite $Q \subseteq \{\{f,g\} \subseteq \mathrm{B}_{\mathcal{P}}(h,4\Delta) : \mathcal{P}(x : f(x) \neq g(x)) \geq \Delta\}$ also has $Q \subseteq \{\{f,g\} \subseteq \mathbb{C} : \mathcal{P}(x : f(x) \neq g(x)) \geq \varepsilon\}$, we have $\bar{\rho}(\varepsilon) \leq \hat{\bar{\rho}}(\varepsilon)$. Thus, it suffices to show $\left\lfloor \frac{1}{\bar{\rho}(\varepsilon)} \right\rfloor \leq \mathfrak{s} \wedge \left\lfloor \frac{1}{\varepsilon} \right\rfloor$.

That $\bar{\rho}(\varepsilon) \geq \varepsilon$ was established by Dasgupta (2005, Lemma 1); we repeat the argument here for completeness. Fix any probability measure $\mathcal{P}$ over $\mathcal{X}$ and any $\varepsilon,\tau \in (0,1]$ with $\tau < \varepsilon$. Fix any finite set $Q \subseteq \{\{f,g\} \subseteq \mathbb{C} : \mathcal{P}(x : f(x) \neq g(x)) \geq \varepsilon\}$. If $Q = \emptyset$, then trivially $\mathcal{P}(x : \mathrm{Split}(Q,x) \geq \varepsilon|Q|) = 1 \geq \tau$. Otherwise, if $Q \neq \emptyset$, letting $X \sim \mathcal{P}$,

$$\mathbb{E}[\mathrm{Split}(Q,X)] \geq \mathbb{E}\left[\sum_{\{f,g\}\in Q} \mathbb{1}[f(Z) \neq g(Z)]\right] = \sum_{\{f,g\}\in Q} \mathcal{P}(x : f(x) \neq g(x)) \geq |Q|\varepsilon.$$

Furthermore, since $\mathrm{Split}(Q,x) \leq |Q|$,

$$\begin{aligned}
&\mathbb{E}[\mathrm{Split}(Q,X)] \\
&= \mathbb{E}\left[\mathbb{1}[\mathrm{Split}(Q,X) \geq (\varepsilon-\tau)|Q|]\mathrm{Split}(Q,X)\right] + \mathbb{E}\left[\mathbb{1}[\mathrm{Split}(Q,X) < (\varepsilon-\tau)|Q|]\mathrm{Split}(Q,X)\right] \\
&< \mathcal{P}\left(x : \mathrm{Split}(Q,x) \geq (\varepsilon-\tau)|Q|\right)|Q| + (\varepsilon-\tau)|Q|.
\end{aligned}$$

Together, these inequalities imply

$$|Q|\varepsilon < \mathcal{P}\left(x : \text{Split}(Q,x) \geq (\varepsilon-\tau)|Q|\right)|Q| + (\varepsilon-\tau)|Q|.$$

Subtracting $(\varepsilon-\tau)|Q|$ from both sides and dividing by $|Q|$, we have

$$\tau < \mathcal{P}\left(x : \text{Split}(Q,x) \geq (\varepsilon-\tau)|Q|\right).$$

Since this holds for any such Q, we have that $\mathbb{C}$ is $((\varepsilon-\tau),\varepsilon,\tau)$-splittable under $\mathcal{P}$, so that $\bar{\rho}_{\mathcal{P}}(\varepsilon;\tau) \geq \varepsilon-\tau$. Since this holds for every choice of $\mathcal{P}$, we have that

$$\bar{\rho}(\varepsilon) = \inf_{\mathcal{P}} \lim_{\tau\to 0} \bar{\rho}_{\mathcal{P}}(\varepsilon;\tau) \geq \lim_{\tau\to 0} \varepsilon - \tau = \varepsilon,$$

from which it immediately follows that $\left\lfloor \frac{1}{\bar{\rho}(\varepsilon)} \right\rfloor \leq \left\lfloor \frac{1}{\varepsilon} \right\rfloor$.

It remains only to show that $\left\lfloor \frac{1}{\bar{\rho}(\varepsilon)} \right\rfloor \leq \mathfrak{s}$. In particular, since this trivially holds when $\mathfrak{s} = \infty$, for the remainder of the proof we suppose $\mathfrak{s} < \infty$. As argued in Section 4, we have $d \leq \mathfrak{s}$, so that this also implies $d < \infty$. Thus, Lemma 43 implies that $\bar{\rho}(\varepsilon) \geq \lim_{\gamma\to 0} \inf_{\mathcal{P}\in\Pi} \lim_{\tau\to 0} \bar{\rho}_{\mathcal{P}}((1-\gamma)\varepsilon;\tau)$. Therefore, if we can establish that, for every $\varepsilon \in (0,1]$ and $\mathcal{P} \in \Pi$, $\lim_{\tau\to 0} \bar{\rho}_{\mathcal{P}}(\varepsilon;\tau) \geq 1/\mathfrak{s}$, then we would have that for every $\varepsilon \in (0,1]$,

$$\left\lfloor \frac{1}{\bar{\rho}(\varepsilon)} \right\rfloor \leq \frac{1}{\bar{\rho}(\varepsilon)} \leq \lim_{\gamma\to 0} \sup_{\mathcal{P}\in\Pi} \frac{1}{\lim_{\tau\to 0} \bar{\rho}_{\mathcal{P}}((1-\gamma)\varepsilon;\tau)} \leq \mathfrak{s},$$

which would thereby complete the proof.

Toward this end, fix any $\varepsilon \in (0,1]$, and for each $\mathcal{P} \in \Pi$, denote $\tau_{\mathcal{P}} = \min\{\mathcal{P}(\{x\}) : x \in \mathcal{X}, \mathcal{P}(\{x\}) > 0\}$; in particular, note that (since $\mathcal{P} \in \Pi$) $0 < \tau_{\mathcal{P}} \leq 1$, and therefore also that, $\forall \varepsilon \in (0,1]$, $\lim_{\tau\to 0} \bar{\rho}_{\mathcal{P}}(\varepsilon;\tau) \geq \bar{\rho}_{\mathcal{P}}(\varepsilon;\tau_{\mathcal{P}})$ (in fact, they are equal). Furthermore, denoting $\text{supp}(\mathcal{P}) = \{x \in \mathcal{X} : \mathcal{P}(\{x\}) > 0\}$, every $x \in \text{supp}(\mathcal{P})$ has $\mathcal{P}(\{x\}) \geq \tau_{\mathcal{P}}$, while $\mathcal{P}(\mathcal{X} \setminus \text{supp}(\mathcal{P})) = 0$. Thus, for any finite $Q \subseteq \{\{f,g\} \subseteq \mathbb{C} : \mathcal{P}(x : f(x) \neq g(x)) \geq \varepsilon\}$, and any $\rho \in [0,1]$, $\mathcal{P}(x : \text{Split}(Q,x) \geq \rho|Q|) \geq \tau_{\mathcal{P}}$ if and only if $\max_{x\in\text{supp}(\mathcal{P})} \text{Split}(Q,x) \geq \rho|Q|$. Furthermore, since $\mathcal{P}(\mathcal{X} \setminus \text{supp}(\mathcal{P})) = 0$, for any $\varepsilon \in (0,1]$, every $\{f,g\} \subseteq \mathbb{C}$ with $\mathcal{P}(x : f(x) \neq g(x)) \geq \varepsilon$ must have $\text{DIS}(\{f,g\}) \cap \text{supp}(\mathcal{P}) \neq \emptyset$. Thus, defining

$$\mathring{\rho}_{\mathcal{P}} = \sup\Bigg\{\rho \in [0,1] : \forall \text{ finite } Q \subseteq \{\{f,g\} \subseteq \mathbb{C} : \text{DIS}(\{f,g\}) \cap \text{supp}(\mathcal{P}) \neq \emptyset\}, \\ \max_{x\in\text{supp}(\mathcal{P})} \text{Split}(Q,x) \geq \rho|Q|\Bigg\},$$

we have $\mathring{\rho}_{\mathcal{P}} \leq \bar{\rho}_{\mathcal{P}}(\varepsilon;\tau_{\mathcal{P}})$ for all $\varepsilon \in (0,1]$ (in fact, they are equal for $\varepsilon \leq \tau_{\mathcal{P}}$). Thus, it suffices to show that $\inf_{\mathcal{P}\in\Pi} \mathring{\rho}_{\mathcal{P}} \geq 1/\mathfrak{s}$. Now partition the set Π by the sizes of the supports, defining, for each $m \in \mathbb{N}$, $\Pi_m = \{\mathcal{P} \in \Pi : |\text{supp}(\mathcal{P})| = m\}$ (this is slightly different from the definition used in the proof of Theorem 10). Note that, for any $\mathcal{P} \in \Pi$, the value of $\mathring{\rho}_{\mathcal{P}}$ is entirely determined by $\text{supp}(\mathcal{P})$. Thus, defining, $\forall m \in \mathbb{N}$ with $m \leq |\mathcal{X}|$,

$$\mathring{\rho}_m = \inf_{\mathcal{X}_m \subseteq \mathcal{X} : |\mathcal{X}_m| = m} \sup\Bigg\{\rho \in [0,1] : \forall \text{ finite } Q \subseteq \{\{f,g\} \subseteq \mathbb{C} : \text{DIS}(\{f,g\}) \cap \mathcal{X}_m \neq \emptyset\}, \\ \max_{x\in\mathcal{X}_m} \text{Split}(Q,x) \geq \rho|Q|\Bigg\},$$

we have $\inf_{\mathcal{P}\in\Pi_m} \mathring{\rho}_{\mathcal{P}} \geq \mathring{\rho}_m$ (in fact, they are equal). Thus, since $\Pi = \bigcup_{m\in\mathbb{N}} \Pi_m$, we have $\inf_{\mathcal{P}\in\Pi} \mathring{\rho}_{\mathcal{P}} = \inf_{m\in\mathbb{N}:m\leq|\mathcal{X}|} \inf_{\mathcal{P}\in\Pi_m} \mathring{\rho}_{\mathcal{P}} \geq \inf_{m\in\mathbb{N}:m\leq|\mathcal{X}|} \mathring{\rho}_m$. Therefore, it suffices to show that $\mathring{\rho}_m \geq 1/\mathfrak{s}$ for all $m \in \mathbb{N}$ with $m \leq |\mathcal{X}|$.

We proceed by induction on $m \in \mathbb{N}$ with $m \leq |\mathcal{X}|$, combined with a nested inductive argument on Q. As base cases (for induction on m), consider any $m \leq \mathfrak{s}$. Fix any $\mathcal{X}_m \subseteq \mathcal{X}$ with $|\mathcal{X}_m| = m$ (noting that $m \leq \mathfrak{s}$ implies $m \leq |\mathcal{X}|$, since $\mathfrak{s} \leq |\mathcal{X}|$ immediately follows from Definition 2). Also fix any finite set $Q \subseteq \{\{f,g\} \subseteq \mathbb{C} : \mathrm{DIS}(\{f,g\}) \cap \mathcal{X}_m \neq \emptyset\}$. Since $\forall \{f,g\} \in Q$, $\exists x \in \mathcal{X}_m$ such that $f(x) \neq g(x)$, the pigeonhole principle implies $\exists x \in \mathcal{X}_m$ with $|\{\{f,g\} \in Q : f(x) \neq g(x)\}| \geq |Q|/|\mathcal{X}_m| = |Q|/m$. For this x, we have $\mathrm{Split}(Q,x) \geq |\{\{f,g\} \in Q : f(x) \neq g(x)\}| \geq (1/m)|Q| \geq (1/\mathfrak{s})|Q|$. Since this holds for any such choice of Q and $\mathcal{X}_m$, we have that $\mathring{\rho}_m \geq 1/\mathfrak{s}$.

If $|\mathcal{X}| = \mathfrak{s}$, this completes the proof. Otherwise, take as an inductive hypothesis that, for some $m \in \mathbb{N}$ with $\mathfrak{s} < m \leq |\mathcal{X}|$, $\mathring{\rho}_{m-1} \geq 1/\mathfrak{s}$. Fix any $\mathcal{X}_m \subseteq \mathcal{X}$ with $|\mathcal{X}_m| = m$. We now introduce a nested inductive argument on Q (based on the partial ordering induced by the subset relation). As a base case, if $Q = \emptyset$, then trivially $\max_{x\in\mathcal{X}_m} \mathrm{Split}(Q,x) = 0 = (1/\mathfrak{s})|Q|$. Now take as a nested inductive hypothesis that, for some nonempty finite set $Q \subseteq \{\{f,g\} \subseteq \mathbb{C} : \mathrm{DIS}(\{f,g\}) \cap \mathcal{X}_m \neq \emptyset\}$, for every strict subset $R \subset Q$, $\max_{x\in\mathcal{X}_m} \mathrm{Split}(R,x) \geq (1/\mathfrak{s})|R|$.

First, consider the case in which $\exists x \in \mathcal{X}_m$ such that $x \notin \bigcup_{\{f,g\}\in Q} \mathrm{DIS}(\{f,g\})$. In this case, every $\{f,g\} \in Q$ has $\mathrm{DIS}(\{f,g\}) \cap (\mathcal{X}_m \setminus \{x\}) = \mathrm{DIS}(\{f,g\}) \cap \mathcal{X}_m \neq \emptyset$, so that $Q \subseteq \{\{f,g\} \subseteq \mathbb{C} : \mathrm{DIS}(\{f,g\}) \cap (\mathcal{X}_m \setminus \{x\}) \neq \emptyset\}$. Therefore, since $|\mathcal{X}_m \setminus \{x\}| = m-1$, by definition of $\mathring{\rho}_{m-1}$ we have $\max_{x'\in\mathcal{X}_m} \mathrm{Split}(Q,x') \geq \max_{x'\in\mathcal{X}_m\setminus\{x\}} \mathrm{Split}(Q,x') \geq \mathring{\rho}_{m-1}|Q|$. Combined with the inductive hypothesis (for m), this implies $\max_{x'\in\mathcal{X}_m} \mathrm{Split}(Q,x') \geq (1/\mathfrak{s})|Q|$.

Now consider the remaining case, in which $\forall x \in \mathcal{X}_m$, $\exists \{f_x, g_x\} \in Q$ with $x \in \mathrm{DIS}(\{f_x, g_x\})$. Since $\{f_x, g_x\} \notin Q_x^y$ for every $y \in \mathcal{Y}$ and $x \in \mathcal{X}_m$, we have $\max_{x\in\mathcal{X}_m} \mathrm{Split}(Q,x) \geq 1$. We proceed by a kind of set-covering argument, as follows. For each $x \in \mathcal{X}_m$, denote $y_x = \mathrm{argmax}_{y\in\mathcal{Y}} |Q_x^y|$ (breaking ties arbitrarily), and denote $S_x = \{x' \in \mathcal{X}_m : \{f_x, g_x\} \notin Q_{x'}^{y_{x'}}\}$. Let z_1 be any element of $\mathcal{X}_m$. Then, for integers $i \geq 2$, inductively define z_i as any element of $\mathcal{X}_m \setminus \bigcup_{j=1}^{i-1} S_{z_j}$, up until the smallest index $i \in \mathbb{N}$ for which $\mathcal{X}_m \setminus \bigcup_{j=1}^{i} S_{z_i} = \emptyset$; denote by I this smallest i with $\mathcal{X}_m \setminus \bigcup_{j=1}^{i} S_{z_i} = \emptyset$. Note that, since $\{f_x, g_x\} \notin Q_x^{y_x}$ (and hence $x \in S_x$) for each $x \in \mathcal{X}_m$, every z_i is distinct, which further implies that $I \leq m$ (and in particular, that I exists). Furthermore, since any $i \in \{1,\ldots,I\}$ and $x \in \mathcal{X}_m$ with $\{f_x, g_x\} = \{f_{z_i}, g_{z_i}\}$ have $S_x = S_{z_i}$, and therefore $x \in S_{z_i}$, $\nexists j > i$ with $z_j = x$. Thus, we also have that $\{f_{z_i}, g_{z_i}\} \neq \{f_{z_j}, g_{z_j}\}$ for every $i, j \in \{1,\ldots,I\}$ with $i \neq j$.

Now let $i_1 = I$, and for integers $k \geq 2$, inductively define

$$i_k = \max\left\{ i \in \{1,\ldots,i_{k-1}-1\} : \left(S_{z_i} \setminus \bigcup_{j=1}^{i-1} S_{z_j} \right) \setminus \bigcup_{j=1}^{k-1} S_{z_{i_j}} \neq \emptyset \right\},$$

up to the smallest index $k \in \mathbb{N}$ with $\left\{ i \in \{1,\ldots,i_k-1\} : \left(S_{z_i} \setminus \bigcup_{j=1}^{i-1} S_{z_j} \right) \setminus \bigcup_{j=1}^{k} S_{z_{i_j}} \neq \emptyset \right\} = \emptyset$; denote by K this final value of k (which must exist, since $i_{k+1} \in \mathbb{N}$ is defined and strictly smaller than i_k for any k for which this set is nonempty; in particular, $1 \leq K \leq I$). Finally, let $x_1 = z_{i_1}$, and for each $k \in \{1,\ldots,K\}$, let x_k denote any element of $\left(S_{z_{i_k}} \setminus \bigcup_{j=1}^{i_k-1} S_{z_j} \right) \setminus \bigcup_{j=1}^{k-1} S_{z_{i_j}}$, which is nonempty by definition of i_k.

We first establish, by induction, that $\bigcup_{k=1}^{K} S_{z_{i_k}} = \mathcal{X}_m$. By construction, we have $\bigcup_{i=1}^{I} S_{z_i} = \mathcal{X}_m$. Furthermore, for any $i \in \{1,\ldots,I\}$, if $\bigcup_{j<i} S_{z_j} \cup \bigcup_{1\le k\le K: i_k \ge i+1} S_{z_{i_k}} = \mathcal{X}_m$, then either $i \in \{i_1,\ldots,i_K\}$, in which case $\bigcup_{j<i} S_{z_j} \cup \bigcup_{1\le k\le K: i_k \ge i} S_{z_{i_k}} = \bigcup_{j\le i} S_{z_j} \cup \bigcup_{1\le k\le K: i_k \ge i+1} S_{z_{i_k}} = \mathcal{X}_m$, or else $i \notin \{i_1,\ldots,i_K\}$, which (by definition of the i_k sequence) implies $S_{z_i} \subseteq \bigcup_{j=1}^{i-1} S_{z_j} \cup \bigcup_{1\le k\le K: i_k \ge i+1} S_{z_{i_k}}$, so that $\bigcup_{j<i} S_{z_j} \cup \bigcup_{1\le k\le K: i_k \ge i} S_{z_{i_k}} = \bigcup_{j<i} S_{z_j} \cup \bigcup_{1\le k\le K: i_k \ge i+1} S_{z_{i_k}} = \bigcup_{j\le i} S_{z_j} \cup \bigcup_{1\le k\le K: i_k \ge i+1} S_{z_{i_k}} = \mathcal{X}_m$. By induction, we have that $\bigcup_{k=1}^{K} S_{z_{i_k}} = \bigcup_{j<1} S_{z_j} \cup \bigcup_{1\le k\le K: i_k \ge 1} S_{z_{i_k}} = \mathcal{X}_m$. In other words, $\forall x \in \mathcal{X}_m$, $\exists k(x) \in \{1,\ldots,K\}$ with $\{f_{z_{i_{k(x)}}}, g_{z_{i_{k(x)}}}\} \notin Q_x^{y_x}$.

In particular, letting $R = Q \setminus \{\{f_{z_{i_k}}, g_{z_{i_k}}\} : k \in \{1,\ldots,K\}\}$, we have that $\forall x \in \mathcal{X}_m$, $\{f_{z_{i_{k(x)}}}, g_{z_{i_{k(x)}}}\} \in (Q \setminus R) \setminus (Q_x^{y_x} \setminus R)$ while $Q_x^{y_x} \setminus R \subseteq Q \setminus R$, so that $|Q \setminus R| - |Q_x^{y_x} \setminus R| \ge 1$. Therefore, $\forall x \in \mathcal{X}_m$,

$$\begin{aligned}
\mathrm{Split}(R,x) &= |R| - \max_{y\in\mathcal{Y}} |R_x^y| \le |R| - |R_x^{y_x}| = |R| - |R \cap Q_x^{y_x}| \\
&= (|Q| - |Q \setminus R|) - (|Q_x^{y_x}| - |Q_x^{y_x} \setminus R|) = (|Q| - |Q_x^{y_x}|) - (|Q \setminus R| - |Q_x^{y_x} \setminus R|) \\
&\le |Q| - |Q_x^{y_x}| - 1 = |Q| - \max_{y\in\mathcal{Y}} |Q_x^y| - 1 = \mathrm{Split}(Q,x) - 1. \qquad (61)
\end{aligned}$$

Since $K \ge 1$, we may note that R is a strict subset of Q, so that the (nested) inductive hypothesis implies that $\max_{x\in\mathcal{X}_m} \mathrm{Split}(R,x) \ge (1/\mathfrak{s})|R|$. Combined with (61), this implies

$$\max_{x\in\mathcal{X}_m} \mathrm{Split}(Q,x) \ge \max_{x\in\mathcal{X}_m} \mathrm{Split}(R,x) + 1 \ge (1/\mathfrak{s})|R| + 1. \qquad (62)$$

Next, we argue that $K \le \mathfrak{s}$, by proving that $\{x_1,\ldots,x_K\}$ is a star set for $\mathbb{C}$. By definition of z_I, we have $z_I \in \mathcal{X}_m \setminus \bigcup_{j=1}^{I-1} S_{z_j} \subseteq \mathcal{X}_m \setminus \bigcup_{k=2}^{K} S_{z_{i_k}}$. Furthermore, $z_I \in S_{z_I}$, so that $z_I \in S_{z_I} \setminus \bigcup_{k=2}^{K} S_{z_{i_k}}$. Since $x_1 = z_{i_1} = z_I$, we have $x_1 \in S_{z_{i_1}} \setminus \bigcup_{k=2}^{K} S_{z_{i_k}}$. Also, for each $k \in \{2,\ldots,K\}$, by definition, $x_k \in \left(S_{z_{i_k}} \setminus \bigcup_{j=1}^{i_k-1} S_{z_j}\right) \setminus \bigcup_{j=1}^{k-1} S_{z_{i_j}} \subseteq \left(S_{z_{i_k}} \setminus \bigcup_{j=k+1}^{K} S_{z_{i_j}}\right) \setminus \bigcup_{j=1}^{k-1} S_{z_{i_j}} = S_{z_{i_k}} \setminus \bigcup_{1\le j\le K: j\ne k} S_{z_{i_j}}$. Therefore, every $k \in \{1,\ldots,K\}$ has $x_k \in S_{z_{i_k}} \setminus \bigcup_{1\le j\le K: j\ne k} S_{z_{i_j}}$. In particular, for every $k \in \{1,\ldots,K\}$, since $x_k \in S_{z_{i_k}}$, we have $\{f_{z_{i_k}}, g_{z_{i_k}}\} \notin Q_{x_k}^{y_{x_k}}$, so that $\exists h_k \in \{f_{z_{i_k}}, g_{z_{i_k}}\}$ with $h_k(x_k) \ne y_{x_k}$. Furthermore, for every $j \in \{1,\ldots,K\} \setminus \{k\}$, since $x_j \notin S_{z_{i_k}}$, we have $\{f_{z_{i_k}}, g_{z_{i_k}}\} \in Q_{x_j}^{y_{x_j}}$, so that $f_{z_{i_k}}(x_j) = g_{z_{i_k}}(x_j) = y_{x_j}$, and in particular, $h_k(x_j) = y_{x_j}$. Also, since we have chosen $x_1 = z_{i_1}$, so that $x_1 \in \mathrm{DIS}(\{f_{z_{i_1}}, g_{z_{i_1}}\})$, $\exists h_0 \in \{f_{z_{i_1}}, g_{z_{i_1}}\}$ with $h_0(x_1) \ne h_1(x_1)$: that is, $h_0(x_1) = y_{x_1}$. Thus, since $f_{z_{i_1}}(x_j) = g_{z_{i_1}}(x_j) = y_{x_j}$ for every $j \in \{2,\ldots,K\}$, we have that $h_0(x_k) = y_{x_k}$ for every $k \in \{1,\ldots,K\}$. Altogether, we have that every $k \in \{1,\ldots,K\}$ has $h_k(x_k) \ne h_0(x_k)$, while every $j \in \{1,\ldots,K\} \setminus \{k\}$ has $h_k(x_j) = h_0(x_j)$. In other words, $\forall k \in \{1,\ldots,K\}$, $\mathrm{DIS}(\{h_0,h_k\}) \cap \{x_1,\ldots,x_K\} = \{x_k\}$: that is, $\{x_1,\ldots,x_K\}$ is a star set for $\mathbb{C}$, witnessed by $\{h_0,h_1,\ldots,h_K\}$. In particular, this implies $K \le \mathfrak{s}$.

Therefore, since $|Q \setminus R| = K$ (by distinctness of the pairs $\{f_{z_i}, g_{z_i}\}$ argued above), (62) implies

$$\max_{x\in\mathcal{X}_m} \mathrm{Split}(Q,x) \ge (1/\mathfrak{s})|R| + \frac{K}{\mathfrak{s}} = (1/\mathfrak{s})(|R| + |Q \setminus R|) = (1/\mathfrak{s})|Q|.$$

By the principle of induction (on Q), we have $\max_{x\in\mathcal{X}_m} \mathrm{Split}(Q,x) \ge (1/\mathfrak{s})|Q|$ for every finite set $Q \subseteq \{\{f,g\} \subseteq \mathbb{C} : \mathrm{DIS}(\{f,g\}) \cap \mathcal{X}_m \ne \emptyset\}$. Since this holds for any choice of

$\mathcal{X}_m$ with $|\mathcal{X}_m| = m$, we have $\mathring{\rho}_m \geq 1/\mathfrak{s}$. By the principle of induction (on m), we have established that $\mathring{\rho}_m \geq 1/\mathfrak{s}$ for every $m \in \mathbb{N}$ with $m \leq |\mathcal{X}|$, which completes the proof of the theorem. ■

C.3 The Teaching Dimension

Here we give the proofs of results from Section 7.3. We first prove that every minimal specifying set is a star set (Lemma 14). In fact, we establish a slightly stronger claim here (which also applies to local minima), stated formally as follows.

Lemma 44 *Fix any $h : \mathcal{X} \to \mathcal{Y}$, $m \in \mathbb{N}$, $\mathcal{U} \in \mathcal{X}^m$, and any specifying set S for h on $\mathcal{U}$ with respect to $\mathbb{C}[\mathcal{U}]$. If $\forall x \in S$, $S \setminus \{x\}$ is not a specifying set for h on $\mathcal{U}$ with respect to $\mathbb{C}[\mathcal{U}]$, then S is a star set for $\mathbb{C} \cup \{h\}$ centered at h.*

Proof Fix an arbitrary sequence $\mathcal{U} = \{x_1, \ldots, x_m\} \in \mathcal{X}^m$ and any $h : \mathcal{X} \to \mathcal{Y}$. Let $t \geq \mathrm{TD}(h, \mathbb{C}[\mathcal{U}], \mathcal{U})$, and let $i_1, \ldots, i_t \in \{1, \ldots, m\}$ be such that $S = \{x_{i_1}, \ldots, x_{i_t}\}$ is a specifying set for h on $\mathcal{U}$ with respect to $\mathbb{C}[\mathcal{U}]$. First note that, if $\exists j \in \{1, \ldots, t\}$ such that every $g \in V_{S\setminus\{x_{i_j}\},h}$ has $g(x_{i_j}) = h(x_{i_j})$ (which includes the case $V_{S\setminus\{x_{i_j}\},h} = \emptyset$), then $V_{S\setminus\{x_{i_j}\},h} = V_{S,h}$, so that $|V_{S\setminus\{x_{i_j}\},h} \cap \mathbb{C}[\mathcal{U}]| = |V_{S,h} \cap \mathbb{C}[\mathcal{U}]| \leq 1$; thus, $S \setminus \{x_{i_j}\}$ is also a specifying set for h on $\mathcal{U}$ with respect to $\mathbb{C}[\mathcal{U}]$.

Therefore, if S is such that $\forall j \leq t$, $S \setminus \{x_{i_j}\}$ is *not* a specifying set for h on $\mathcal{U}$ with respect to $\mathbb{C}[\mathcal{U}]$, then $\forall j \in \{1, \ldots, t\}$, $\exists h_j \in V_{S\setminus\{x_{i_j}\},h}$ with $h_j(x_{i_j}) \neq h(x_{i_j})$; noting that "$h_j \in V_{S\setminus\{x_{i_j}\},h}$" is equivalent to saying "$h_j(x_{i_k}) = h(x_{i_k})$ for every $k \in \{1, \ldots, t\} \setminus \{j\}$," this precisely matches the definition of a star set in Section 4: that is, we have proven that $\{x_{i_1}, \ldots, x_{i_t}\}$ is a star set for $\mathbb{C} \cup \{h\}$, witnessed by $\{h, h_1, \ldots, h_t\}$, and hence centered at h. ■

Proof of Lemma 14 Lemma 14 follows immediately from Lemma 44 by noting that, for any *minimal* specifying set S for h on $\mathcal{U}$ with respect to $\mathbb{C}[\mathcal{U}]$, $\forall x \in S$, $|S \setminus \{x\}| < \mathrm{TD}(h, \mathbb{C}[\mathcal{U}], \mathcal{U})$, so that $S \setminus \{x\}$ cannot possibly be a specifying set for h on $\mathcal{U}$ with respect to $\mathbb{C}[\mathcal{U}]$. ■

We are now ready for the proof of Theorem 13.

Proof of Theorem 13 Fix any $m \in \mathbb{N}$. First, note that for $\{x_i\}_{i=1}^{\mathfrak{s}}$ and $\{h_i\}_{i=0}^{\mathfrak{s}}$ as in Definition 2, letting $\mathcal{U} = \{x_1, \ldots, x_{\min\{\mathfrak{s},m\}}\}$, for any positive integer $i \leq \min\{\mathfrak{s}, m\}$, any subsequence $S \subseteq \mathcal{U}$ with $x_i \notin S$ has $\{h_0, h_i\} \subseteq V_{S,h_0}$. Thus, since $x_i \in \mathcal{U}$, and $h_0(x_i) \neq h_i(x_i)$, we have $|V_{S,h_0} \cap \mathbb{C}[\mathcal{U}]| \geq 2$. Since this is true for every such $i \leq \min\{\mathfrak{s}, m\}$, every $S \subseteq \mathcal{U}$ without $\{x_1, \ldots, x_{\min\{\mathfrak{s},m\}}\} \subseteq S$ has $|V_{S,h_0} \cap \mathbb{C}[\mathcal{U}]| \geq 2$. Therefore, $\mathrm{TD}(h_0, \mathbb{C}[\mathcal{U}], \mathcal{U}) \geq \min\{\mathfrak{s}, m\}$. Thus, by the definitions of XTD and TD, monotonicity of maximization in the set maximized over, and monotonicity of $t \mapsto \mathrm{TD}(\mathbb{C}, t)$,[15] we have

$$\mathrm{XTD}(\mathbb{C}, m) \geq \mathrm{TD}(\mathbb{C}, m) \geq \mathrm{TD}(\mathbb{C}, \min\{\mathfrak{s}, m\}) \geq \mathrm{TD}(h_0, \mathbb{C}[\mathcal{U}], \mathcal{U}) \geq \min\{\mathfrak{s}, m\}.$$

15. $\forall S \in \mathcal{X}^t$, $\forall x \in S$, $\forall h$, $\mathrm{TD}(h, \mathbb{C}[S \cup \{x\}], S \cup \{x\}) = \mathrm{TD}(h, \mathbb{C}[S], S)$. Thus, $\mathrm{TD}(\mathbb{C}, t+1) = \max_{h \in \mathbb{C}} \max_{S \in \mathcal{X}^t} \max_{x \in \mathcal{X}} \mathrm{TD}(h, \mathbb{C}[S \cup \{x\}], S \cup \{x\}) \geq \max_{h \in \mathbb{C}} \max_{S \in \mathcal{X}^t} \max_{x \in S} \mathrm{TD}(h, \mathbb{C}[S \cup \{x\}], S \cup \{x\}) = \max_{h \in \mathbb{C}} \max_{S \in \mathcal{X}^t} \mathrm{TD}(h, \mathbb{C}[S], S) = \mathrm{TD}(\mathbb{C}, t)$.

Furthermore, it follows immediately from the definition that $\mathrm{XTD}(\mathbb{C}, m) \leq m$. Note that this completes the proof in the case that $\mathfrak{s} \geq m$. To address the remaining case, for the remainder of the proof, we suppose $\mathfrak{s} \leq m$, and focus on establishing $\mathrm{XTD}(\mathbb{C}, m) \leq \mathfrak{s}$.

For this, we proceed by induction on m, taking as a base case the fact that $\mathrm{XTD}(\mathbb{C}, \mathfrak{s}) \leq \mathfrak{s}$, which trivially follows from the definition of XTD. Now take as an inductive hypothesis that for some $m > \mathfrak{s}$, we have $\mathrm{XTD}(\mathbb{C}, m-1) \leq \mathfrak{s}$. Fix any sequence $\mathcal{U}_m = \{x_1, \ldots, x_m\} \in \mathcal{X}^m$, and $h : \mathcal{X} \to \mathcal{Y}$, and denote $\mathcal{U}_{m-1} = \{x_1, \ldots, x_{m-1}\}$. Let $t \in \mathbb{N} \cup \{0\}$ and $S \in \mathcal{U}_{m-1}^t$ be such that S is a minimal specifying set for h on $\mathcal{U}_{m-1}$ with respect to $\mathbb{C}[\mathcal{U}_{m-1}]$. If $|S| \geq \mathrm{TD}(h, \mathbb{C}[\mathcal{U}_m], \mathcal{U}_m)$, then since S is a *minimal* specifying set for h on $\mathcal{U}_{m-1}$ with respect to $\mathbb{C}[\mathcal{U}_{m-1}]$, we have $|S| = \mathrm{TD}(h, \mathbb{C}[\mathcal{U}_{m-1}], \mathcal{U}_{m-1}) \leq \mathrm{XTD}(\mathbb{C}, m-1) \leq \mathfrak{s}$ by the inductive hypothesis; thus, in this case we have $\mathrm{TD}(h, \mathbb{C}[\mathcal{U}_m], \mathcal{U}_m) \leq |S| \leq \mathfrak{s}$. On the other hand, suppose $|S| < \mathrm{TD}(h, \mathbb{C}[\mathcal{U}_m], \mathcal{U}_m)$. In this case, since S is a specifying set for h on $\mathcal{U}_{m-1}$ with respect to $\mathbb{C}[\mathcal{U}_{m-1}]$, we have $\mathrm{DIS}(V_{S,h}) \cap \mathcal{U}_m \subseteq (\mathrm{DIS}(V_{S,h}) \cap \mathcal{U}_{m-1}) \cup \{x_m\} = \{x_m\}$. But since $|S| < \mathrm{TD}(h, \mathbb{C}[\mathcal{U}_m], \mathcal{U}_m)$, S cannot be a specifying set for h on $\mathcal{U}_m$ with respect to $\mathbb{C}[\mathcal{U}_m]$, so that $\mathrm{DIS}(V_{S,h}) \cap \mathcal{U}_m \neq \emptyset$. Therefore, $\mathrm{DIS}(V_{S,h}) \cap \mathcal{U}_m = \{x_m\}$. In particular, this implies that $S \cup \{x_m\}$ is a specifying set for h on $\mathcal{U}_m$ with respect to $\mathbb{C}[\mathcal{U}_m]$, and in particular, must be a *minimal* such specifying set, since $|S \cup \{x_m\}| = |S| + 1 \leq \mathrm{TD}(h, \mathbb{C}[\mathcal{U}_m], \mathcal{U}_m)$. Therefore, Lemma 14 implies that $S \cup \{x_m\}$ is a star set for $\mathbb{C} \cup \{h\}$ centered at h. If $h \in \mathbb{C}$, this already implies that $|S \cup \{x_m\}| \leq \mathfrak{s}$; furthermore, we can argue that this remains the case even if $h \notin \mathbb{C}$, as follows. Since $x_m \in \mathrm{DIS}(V_{S,h})$, we have $V_{S \cup \{x_m\}, h} \neq \emptyset$, so that $\exists g_0 \in \mathbb{C}$ such that $\forall x \in S \cup \{x_m\}$, $g_0(x) = h(x)$. Therefore, $S \cup \{x_m\}$ is also a star set for $\mathbb{C}$ centered at g_0, so that $|S \cup \{x_m\}| \leq \mathfrak{s}$. In particular, since $S \cup \{x_m\}$ is a minimal specifying set for h on $\mathcal{U}_m$ with respect to $\mathbb{C}[\mathcal{U}_m]$, we have $|S \cup \{x_m\}| = \mathrm{TD}(h, \mathbb{C}[\mathcal{U}_m], \mathcal{U}_m)$, so that $\mathrm{TD}(h, \mathbb{C}[\mathcal{U}_m], \mathcal{U}_m) \leq \mathfrak{s}$ in this case as well. Thus, in either case, we have $\mathrm{TD}(h, \mathbb{C}[\mathcal{U}_m], \mathcal{U}_m) \leq \mathfrak{s}$. Maximizing over the choice of h and $\{x_1, \ldots, x_m\}$, we have $\mathrm{XTD}(\mathbb{C}, m) \leq \mathfrak{s}$, which completes the inductive step. The result now follows by the principle of induction. ■

Next, we prove Theorem 15.

Proof of Theorem 15 Fix any $m \in \mathbb{N}$ and $\delta \in [0, 1]$. Let $\{x_i\}_{i=1}^{\mathfrak{s}}$ and $\{h_i\}_{i=0}^{\mathfrak{s}}$ be as in Definition 2, and let $\mathcal{U} = \{x_1, \ldots, x_{\min\{\mathfrak{s}, m\}}\}$ and $\mathcal{G} = \{h_i : i \in \{0, \ldots, \min\{\mathfrak{s}, m\}\}\}$. As in the proof of Theorem 13, for any positive integer $i \leq \min\{\mathfrak{s}, m\}$, any subsequence $S \subseteq \mathcal{U}$ with $x_i \notin S$ has $\{h_0, h_i\} \subseteq V_{S,h_0}$. Thus, since $x_i \in \mathcal{U}$ for every $i \leq \min\{\mathfrak{s}, m\}$, and every h_i realizes a distinct classification of $\mathcal{U}$ ($i \leq \min\{\mathfrak{s}, m\}$), we have $|V_{S,h_0} \cap \mathcal{G}[\mathcal{U}]| \geq |\{i \in \{1, \ldots, \min\{\mathfrak{s}, m\}\} : x_i \notin S\}| + 1 \geq \min\{\mathfrak{s}, m\} - |S| + 1$. In particular, to have $|V_{S,h_0} \cap \mathcal{G}[\mathcal{U}]| \leq \delta |\mathcal{G}[\mathcal{U}]| + 1 = \delta(\min\{\mathfrak{s}, m\} + 1) + 1$, we must have $|S| \geq (1-\delta)\min\{\mathfrak{s}, m\} - \delta$. Therefore, $\mathrm{XPTD}(h_0, \mathcal{G}[\mathcal{U}], \mathcal{U}, \delta) \geq (1-\delta)\min\{\mathfrak{s}, m\} - \delta$. By definition of $\mathrm{XPTD}(\mathcal{H}, m, \delta)$ and the fact that $\mathcal{G} \subseteq \mathbb{C}$, and since $t \mapsto \mathrm{XPTD}(\mathcal{H}, t, \delta)$ is nondecreasing (since $\forall S \in \mathcal{X}^t$, $\forall x \in S$, $\forall h$, $\mathrm{XPTD}(h, \mathcal{H}[S \cup \{x\}], S \cup \{x\}, \delta) = \mathrm{XPTD}(h, \mathcal{H}[S], S, \delta)$), this further implies

$$\begin{aligned}\max_{\mathcal{H} \subseteq \mathbb{C}} \mathrm{XPTD}(\mathcal{H}, m, \delta) &\geq \mathrm{XPTD}(\mathcal{G}, m, \delta) \geq \mathrm{XPTD}(\mathcal{G}, \min\{\mathfrak{s}, m\}, \delta) \\ &\geq \mathrm{XPTD}(h_0, \mathcal{G}[\mathcal{U}], \mathcal{U}, \delta) \geq (1-\delta)\min\{\mathfrak{s}, m\} - \delta \geq (1-2\delta)\min\{\mathfrak{s}, m\},\end{aligned}$$

where this last inequality is due to the assumption that $|\mathbb{C}| \geq 3$ (Section 2), which implies $\mathfrak{s} \geq 1$. Since $\mathrm{XPTD}(\cdot, m, \delta) \in \mathbb{N} \cup \{0\}$, this further implies $\max_{\mathcal{H} \subseteq \mathbb{C}} \mathrm{XPTD}(\mathcal{H}, m, \delta) \geq \lceil (1-2\delta)\min\{\mathfrak{s}, m\} \rceil$ when $\delta \leq 1/2$.

To establish the right inequality, fix any $\mathcal{H} \subseteq \mathbb{C}$, let $\mathcal{U} \in \mathcal{X}^m$ and $h : \mathcal{X} \to \mathcal{Y}$ be such that $\mathrm{XPTD}(h, \mathcal{H}[\mathcal{U}], \mathcal{U}, \delta) = \mathrm{XPTD}(\mathcal{H}, m, \delta)$, and let $S \subseteq \mathcal{U}$ be a minimal specifying set for h on $\mathcal{U}$ with respect to $\mathcal{H}[\mathcal{U}]$. If $\delta = 0$ or $|S| < \frac{1+\delta}{\delta}$, then $|S| - 1 < \left(1 - \frac{\delta}{1+\delta}\right)|S| \leq |S|$, so that $\mathrm{XPTD}(h, \mathcal{H}[\mathcal{U}], \mathcal{U}, \delta) \leq |S| = \left\lceil \left(1 - \frac{\delta}{1+\delta}\right)|S| \right\rceil$. Otherwise, suppose $\delta > 0$ and $|S| \geq \frac{1+\delta}{\delta}$, and let $k = \left\lfloor |S| / \left\lfloor \frac{\delta}{1+\delta}|S| \right\rfloor \right\rfloor$, and note that $k \geq 1$. Let $R_1, \ldots, R_k$ denote disjoint subsequences of S with each $|R_i| = \left\lfloor \frac{\delta}{1+\delta}|S| \right\rfloor$, which must exist since minimality of S guarantees that its elements are distinct. Note that, for each $i \in \{1, \ldots, k\}$, $(V_{S\setminus R_i,h} \setminus V_{S,h}) \cap \mathcal{H}[\mathcal{U}]$ is the set of classifiers g in $\mathcal{H}[\mathcal{U}]$ with $\mathrm{DIS}(\{g,h\}) \cap (S \setminus R_i) = \emptyset$ but $\mathrm{DIS}(\{g,h\}) \cap R_i \neq \emptyset$; in particular, for any $i, j \in \{1, \ldots, k\}$ with $i \neq j$, since $R_j \subseteq S \setminus R_i$ and $R_i \subseteq S \setminus R_j$, $(V_{S\setminus R_i,h} \setminus V_{S,h}) \cap \mathcal{H}[\mathcal{U}]$ and $(V_{S\setminus R_j,h} \setminus V_{S,h}) \cap \mathcal{H}[\mathcal{U}]$ are disjoint. Thus, since $\mathcal{H}[\mathcal{U}] \supseteq (V_{S,h} \cap \mathcal{H}[\mathcal{U}]) \cup \bigcup_{i=1}^{k} (V_{S\setminus R_i,h} \setminus V_{S,h}) \cap \mathcal{H}[\mathcal{U}]$, we have

$$
\begin{aligned}
|\mathcal{H}[\mathcal{U}]| &\geq \left| (V_{S,h} \cap \mathcal{H}[\mathcal{U}]) \cup \bigcup_{i=1}^{k} (V_{S\setminus R_i,h} \setminus V_{S,h}) \cap \mathcal{H}[\mathcal{U}] \right| \\
&= |V_{S,h} \cap \mathcal{H}[\mathcal{U}]| + \sum_{i=1}^{k} \left| (V_{S\setminus R_i,h} \setminus V_{S,h}) \cap \mathcal{H}[\mathcal{U}] \right| \geq \sum_{i=1}^{k} \left| (V_{S\setminus R_i,h} \setminus V_{S,h}) \cap \mathcal{H}[\mathcal{U}] \right| \\
&\geq k \min_{i \in \{1,\ldots,k\}} \left| (V_{S\setminus R_i,h} \setminus V_{S,h}) \cap \mathcal{H}[\mathcal{U}] \right|.
\end{aligned}
$$

Thus, letting $i^* = \mathrm{argmin}_{i \in \{1,\ldots,k\}} \left| (V_{S\setminus R_i,h} \setminus V_{S,h}) \cap \mathcal{H}[\mathcal{U}] \right|$, we have $\left| (V_{S\setminus R_{i^*},h} \setminus V_{S,h}) \cap \mathcal{H}[\mathcal{U}] \right| \leq \frac{1}{k}|\mathcal{H}[\mathcal{U}]|$. Furthermore, since S is a specifying set for h on $\mathcal{U}$ with respect to $\mathcal{H}[\mathcal{U}]$, $|V_{S,h} \cap \mathcal{H}[\mathcal{U}]| \leq 1$, so that (since $V_{S,h} \subseteq V_{S\setminus R_{i^*},h}$)

$$
\begin{aligned}
\left| V_{S\setminus R_{i^*},h} \cap \mathcal{H}[\mathcal{U}] \right| &= \left| \left(\left(V_{S\setminus R_{i^*},h} \setminus V_{S,h} \right) \cap \mathcal{H}[\mathcal{U}] \right) \cup (V_{S,h} \cap \mathcal{H}[\mathcal{U}]) \right| \\
&= \left| \left(V_{S\setminus R_{i^*},h} \setminus V_{S,h} \right) \cap \mathcal{H}[\mathcal{U}] \right| + |V_{S,h} \cap \mathcal{H}[\mathcal{U}]| \leq \frac{1}{k}|\mathcal{H}[\mathcal{U}]| + 1.
\end{aligned}
$$

Also, since

$$
\frac{1}{k} \leq \frac{1}{\left\lfloor \frac{1+\delta}{\delta} \right\rfloor} \leq \frac{1}{\frac{1+\delta}{\delta} - 1} = \delta,
$$

this implies $|V_{S\setminus R_{i^*},h} \cap \mathcal{H}[\mathcal{U}]| \leq \delta|\mathcal{H}[\mathcal{U}]| + 1$, so that $\mathrm{XPTD}(h, \mathcal{H}[\mathcal{U}], \mathcal{U}, \delta) \leq |S \setminus R_{i^*}|$. Furthermore, since $R_{i^*} \subseteq S$, $|S \setminus R_{i^*}| = |S| - |R_{i^*}| = |S| - \left\lfloor \frac{\delta}{1+\delta}|S| \right\rfloor = \left\lceil \left(1 - \frac{\delta}{1+\delta}\right)|S| \right\rceil$.

Thus, for any $\delta \in [0,1]$ and regardless of the size of $|S|$, we have $\mathrm{XPTD}(h, \mathcal{H}[\mathcal{U}], \mathcal{U}, \delta) \leq \left\lceil \left(1 - \frac{\delta}{1+\delta}\right)|S| \right\rceil$. Furthermore, since S is a minimal specifying set for h on $\mathcal{U}$ with respect to $\mathcal{H}[\mathcal{U}]$, we have $|S| \leq \mathrm{XTD}(\mathcal{H}, m) \leq \mathrm{XTD}(\mathbb{C}, m)$, and Theorem 13 implies $\mathrm{XTD}(\mathbb{C}, m) = \min\{\mathfrak{s}, m\}$. Therefore, $\mathrm{XPTD}(h, \mathcal{H}[\mathcal{U}], \mathcal{U}, \delta) \leq \left\lceil \left(1 - \frac{\delta}{1+\delta}\right) \min\{\mathfrak{s}, m\} \right\rceil$. Maximizing the left hand side over the choice of h, $\mathcal{H}$, and $\mathcal{U}$ completes the proof. ■

C.4 The Doubling Dimension

We now present the proof of Theorem 17

Proof of Theorem 17 For the lower bound, fix any $\varepsilon \in (0,1]$, and take $\{x_i\}_{i=1}^{\mathfrak{s}}$ and $\{h_i\}_{i=0}^{\mathfrak{s}}$ as in Definition 2, and let $m = \mathfrak{s} \wedge \left\lfloor \frac{1}{\varepsilon} \right\rfloor$. Let $\mathcal{P}$ be a probability measure on $\mathcal{X}$ with $\mathcal{P}(\{x_i\}) = 1/m$ for each $i \in \{1,\ldots,m\}$. Thus, $\{h_0, h_1, \ldots, h_m\} \subseteq \mathrm{B}_{\mathcal{P}}(h_0, 1/m)$. Furthermore, for any $i \in \{0,\ldots,m\}$ and any classifier g with $\mathcal{P}(x : g(x) \neq h_i(x)) \leq 1/(2m)$, we must have $g(x_j) = h_i(x_j)$ for every $j \in \{1,\ldots,m\}$. Therefore, any $\frac{1}{2m}$-cover of $\mathrm{B}_{\mathcal{P}}(h_0, 1/m)$ must contain classifiers $g_0, \ldots, g_m$ with $\forall i \in \{0,\ldots,m\}$, $\forall j \in \{1,\ldots,m\}$, $g_i(x_j) = h_i(x_j)$. Thus, since each h_i (with $i \leq m$) realizes a distinct classification of $\{x_1,\ldots,x_m\}$, it follows that $\mathcal{N}(1/(2m), \mathrm{B}_{\mathcal{P}}(h_0, 1/m), \mathcal{P}) \geq m+1$. Noting that $1/m \geq \varepsilon$, we have that

$$\sup_P \sup_{h \in \mathbb{C}} D_{h,P}(\varepsilon) \geq D_{h_0,\mathcal{P}}(\varepsilon) \geq \log_2\left(\mathcal{N}\left(\frac{1}{2m}, \mathrm{B}_{\mathcal{P}}\left(h_0, \frac{1}{m}\right), \mathcal{P}\right)\right) \geq \log_2(m+1) \geq \log_2\left(\mathfrak{s} \wedge \frac{1}{\varepsilon}\right).$$

For the remaining term in the lower bound (i.e., d), we modify an argument of Kulkarni (1989, Proposition 3). If $d < 5$, then $d \lesssim \mathrm{Log}\left(\mathfrak{s} \wedge \frac{1}{\varepsilon}\right)$, so that the lower bound follows from the above. Otherwise, suppose $d \geq 5$. We first let $\{x_1', \ldots, x_d'\}$ denote a set of d points in $\mathcal{X}$ shattered by $\mathbb{C}$, and we let G denote the set of classifiers $g \in \mathbb{C}[\{x_1', \ldots, x_d'\}]$ with $g(x_d') = -1$ and $\sum_{i=1}^{d-1} \mathbb{1}[g(x_i') = +1] = \left\lfloor \frac{d-1}{4} \right\rfloor$. For any $g \in G$, note that, if H is a classifier sampled uniformly at random from G, a Chernoff bound (for sampling without replacement) implies

$$\mathbb{P}\left(\sum_{i=1}^{d-1} \mathbb{1}[H(x_i') = g(x_i')] \geq \frac{d-1}{8}\right) \leq \exp\left\{-\frac{d-1}{48}\right\}.$$

Thus, there are at most $|G| \exp\left\{-\frac{d-1}{48}\right\}$ elements $h \in G$ with $\sum_{i=1}^{d-1} \mathbb{1}[h(x_i') = g(x_i')] \geq \frac{d-1}{8}$. Now take $\mathcal{H}_0 = \{\}$, and take as an inductive hypothesis that, for some positive integer $k < 1 + \exp\left\{\frac{d-1}{48}\right\}$, there is a set $\mathcal{H}_{k-1} \subseteq G$ with $|\mathcal{H}_{k-1}| = k-1$ such that $\forall h, g \in \mathcal{H}_{k-1}$ with $h \neq g$, $\sum_{i=1}^{d-1} \mathbb{1}[h(x_i') = g(x_i')] < \frac{d-1}{8}$. Since $|\mathcal{H}_{k-1}| \cdot |G| \exp\left\{-\frac{d-1}{48}\right\} < |G|$, $\exists g_k \in G$ such that $\forall h \in \mathcal{H}_{k-1}$, $\sum_{i=1}^{d-1} \mathbb{1}[h(x_i') = g_k(x_i')] < \frac{d-1}{8}$. Thus, defining $\mathcal{H}_k = \mathcal{H}_{k-1} \cup \{g_k\}$ extends the inductive hypothesis. By induction, this establishes the existence of a set $\mathcal{H} \subseteq G$ with $|\mathcal{H}| \geq \exp\left\{\frac{d-1}{48}\right\}$ such that $\forall h, g \in \mathcal{H}$ with $h \neq g$, $\sum_{i=1}^{d-1} \mathbb{1}[h(x_i') = g(x_i')] < \frac{d-1}{8}$. Fix any $\varepsilon \in (0, 1/4]$ and let $\mathcal{P}$ denote a probability measure over $\mathcal{X}$ with $\mathcal{P}(\{x_i'\}) = \frac{4\varepsilon}{d-1}$ for each $i \in \{1,\ldots,d-1\}$, and $\mathcal{P}(\{x_d'\}) = 1 - 4\varepsilon$. Note that any $h, g \in G$ with $\sum_{i=1}^{d-1} \mathbb{1}[h(x_i') = g(x_i')] < \frac{d-1}{8}$ have $\mathcal{P}(x : h(x) \neq g(x)) > \frac{d-1}{4}\frac{4\varepsilon}{d-1} = \varepsilon$. Thus, $\mathcal{H}$ is an *ε-packing* under the $L_1(\mathcal{P})$ pseudometric. Recall that this implies $|\mathcal{H}| \leq \mathcal{N}(\varepsilon/2, G, \mathcal{P})$ (Kolmogorov and Tikhomirov, 1959, 1961). Furthermore, note that any $g \in G$ has $\mathcal{P}(x : g(x) = +1) = \left\lfloor \frac{d-1}{4} \right\rfloor \frac{4\varepsilon}{d-1} \leq \varepsilon$. Thus, letting $h_- \in \mathbb{C}$ be such that $\forall i \in \{1,\ldots,d\}$, $h_-(x_i') = -1$ (which exists, by shatterability of $x_1', \ldots, x_d'$), we have $G \subseteq \mathrm{B}_{\mathcal{P}}(h_-, \varepsilon)$. Therefore, $\mathcal{N}(\varepsilon/2, G, \mathcal{P}) \leq \mathcal{N}(\varepsilon/2, \mathrm{B}_{\mathcal{P}}(h_-, \varepsilon), \mathcal{P})$. Altogether, we have that

$$d \lesssim \frac{d-1}{48} \log_2(e) \leq \log_2(|\mathcal{H}|) \leq \log_2\left(\mathcal{N}(\varepsilon/2, \mathrm{B}_{\mathcal{P}}(h_-, \varepsilon), \mathcal{P})\right) \leq D_{h_-,\mathcal{P}}(\varepsilon) \leq \sup_P \sup_{h \in \mathbb{C}} D_{h,P}(\varepsilon).$$

For the upper bound, fix any $h \in \mathbb{C}$, any probability measure $\mathcal{P}$ over $\mathcal{X}$, and any $\varepsilon \in (0,1]$, and fix any value $r \in [\varepsilon, 1]$. Recall that any maximal subset $G_r \subseteq \mathrm{B}_{\mathcal{P}}(h, r)$ of classifiers in $\mathrm{B}_{\mathcal{P}}(h, r)$ with $\min_{f,g \in G_r : f \neq g} \mathcal{P}(x : f(x) \neq g(x)) > r/2$ (called a maximal $(r/2)$-packing of $\mathrm{B}_{\mathcal{P}}(h, r)$) is also an $(r/2)$-cover of $\mathrm{B}_{\mathcal{P}}(h, r)$ (see e.g., Kolmogorov and Tikhomirov,

1959, 1961). Thus, we have that $\mathcal{N}\left(\frac{r}{2}, \mathrm{B}_{\mathcal{P}}(h,r), \mathcal{P}\right) \leq |G_r|$, for any such set G_r. Let $m = \left\lceil \frac{4}{r}\ln(|G_r|)\right\rceil$, and let $X_1, X_2, \ldots, X_m$ be independent $\mathcal{P}$-distributed random variables. Let E_1 denote the event that $\forall f, g \in G_r$ with $f \neq g$, $\exists i \in \{1,\ldots,m\}$ with $f(X_i) \neq g(X_i)$. For any $f, g \in G_r$ with $f \neq g$, $\mathbb{P}(\exists i \in \{1,\ldots,m\} : f(X_i) \neq g(X_i)) = 1 - (1 - \mathcal{P}(x : f(x) \neq g(x)))^m > 1 - (1 - r/2)^m > 1 - e^{-mr/2} \geq 1 - 1/|G_r|^2$. Therefore, by a union bound, $\mathbb{P}(E_1) > 1 - \binom{|G_r|}{2}\frac{1}{|G_r|^2} \geq \frac{1}{2}$. In particular, note that on the event E_1, the elements of G_r realize distinct classifications of the sequence $(X_1, \ldots, X_m)$, so that (since $G_r \subseteq \mathrm{B}_{\mathcal{P}}(h,r)$) $|G_r|$ is upper bounded by the number of distinct classifications of $(X_1, \ldots, X_m)$ realized by classifiers in $\mathrm{B}_{\mathcal{P}}(h,r)$. Furthermore, since all classifiers in $\mathrm{B}_{\mathcal{P}}(h,r)$ agree on the classification of any points $X_i \notin \mathrm{DIS}(\mathrm{B}_{\mathcal{P}}(h,r))$, and $\mathrm{B}_{\mathcal{P}}(h,r) \subseteq \mathbb{C}$, we have that $|G_r|$ is upper bounded by the number of distinct classifications of $\{X_1, \ldots, X_m\} \cap \mathrm{DIS}(\mathrm{B}_{\mathcal{P}}(h,r))$ realized by classifiers in $\mathbb{C}$.

By a Chernoff bound, on an event E_2 of probability at least $1/2$,

$$|\{X_1, \ldots, X_m\} \cap \mathrm{DIS}(\mathrm{B}_{\mathcal{P}}(h,r))| \leq 1 + 2e\mathcal{P}(\mathrm{DIS}(\mathrm{B}_{\mathcal{P}}(h,r)))m.$$

By the definition of the disagreement coefficient, this is at most $1 + 2e\theta_{h,\mathcal{P}}(r)rm \leq 1 + 2e + 8e\theta_{h,\mathcal{P}}(r)\ln(|G_r|)$, which, if $|G_r| \geq 3$, is at most $11e\theta_{h,\mathcal{P}}(r)\ln(|G_r|)$. By a union bound, the event $E_1 \cap E_2$ has probability strictly greater than 0. Thus, letting $m' = \lceil 11e\theta_{h,\mathcal{P}}(r)\ln(|G_r|)\rceil$, there exists a sequence $x_1, \ldots, x_{m'} \in \mathcal{X}$ such that $|G_r|$ is at most the max of 2 and the number of distinct classifications of $\{x_1, \ldots, x_{m'}\}$ realized by classifiers in $\mathbb{C}$. In the case $|G_r| \geq 3$, this latter value is at most $\left(\frac{em'}{d}\right)^d \leq \left(\frac{22e^2\theta_{h,\mathcal{P}}(r)\ln(|G_r|)}{d}\right)^d$ by the VC-Sauer lemma (Vapnik and Chervonenkis, 1971; Sauer, 1972).

Taking the logarithm, we have that

$$\ln(|G_r|) \leq \max\left\{\ln(2), d\ln\left(22e^2\theta_{h,\mathcal{P}}(r)\right) + d\ln\left(\frac{\ln(|G_r|)}{d}\right)\right\},$$

which implies (see e.g., Vidyasagar, 2003, Corollary 4.1)

$$\ln(|G_r|) < \max\left\{1, 2d\ln\left(22e^2\theta_{h,\mathcal{P}}(r)\right)\right\} = 2d\ln\left(22e^2\theta_{h,\mathcal{P}}(r)\right).$$

Dividing both sides by $\ln(2)$, altogether we have that

$$\begin{aligned} D_{h,\mathcal{P}}(\varepsilon) &= \sup_{r\in[\varepsilon,1]} \log_2\left(\mathcal{N}\left(\frac{r}{2}, \mathrm{B}_{\mathcal{P}}(h,r), \mathcal{P}\right)\right) \leq \sup_{r\in[\varepsilon,1]} \log_2\left(|G_r|\right) \\ &\leq \sup_{r\in[\varepsilon,1]} 2d\log_2\left(22e^2\theta_{h,\mathcal{P}}(r)\right) = 2d\log_2\left(22e^2\theta_{h,\mathcal{P}}(\varepsilon)\right). \end{aligned}$$

In particular, by Theorem 10, this is at most $2d\log_2\left(22e^2\left(\mathfrak{s}\wedge\frac{1}{\varepsilon}\right)\right)$, so that maximizing the left hand side over the choice of $h \in \mathbb{C}$ and $\mathcal{P}$ completes the proof. ∎

Appendix D. Examples Spanning the Gaps

In this section, taking d and $\mathfrak{s}$ as fixed values in $\mathbb{N}$ (with $d \geq 3$ and $\mathfrak{s} \geq 4d$), and taking $\mathcal{X} = \mathbb{N}$, we establish that the upper bounds in Theorems 3, 4, 5, and 7 are all tight

(up to universal constant and logarithmic factors) when we take $\mathbb{C} = \{x \mapsto 2\mathbb{1}_S(x) - 1 : S \subseteq \{1, \ldots, \mathfrak{s}\}, |S| \leq d\}$, and that the lower bounds in these theorems are all tight (up to logarithmic factors) when we take $\mathbb{C} = \{x \mapsto 2\mathbb{1}_S(x) - 1 : S \in 2^{\{1,\ldots,d\}} \cup \{\{i\} : d+1 \leq i \leq \mathfrak{s}\}\}$. One can easily verify that, in both cases, the VC dimension is indeed d, and the star number is indeed $\mathfrak{s}$.

D.1 The Upper Bounds are Sometimes Tight

We begin with the upper bounds. In this case, take

$$\mathbb{C} = \{x \mapsto 2\mathbb{1}_S(x) - 1 : S \subseteq \{1, \ldots, \mathfrak{s}\}, |S| \leq d\}. \tag{63}$$

For this hypothesis class, we argue that the lower bounds can be increased to match the upper bounds (up to logarithmic factors). We begin with a general lemma.

For each $i \in \{1, \ldots, d\}$, let $\mathcal{X}_i = \{\lfloor \mathfrak{s}/d \rfloor (i-1) + 1, \ldots, \lfloor \mathfrak{s}/d \rfloor i\}$, $\mathbb{C}_i = \{x \mapsto 2\mathbb{1}_{\{t\}}(x) - 1 : t \in \mathcal{X}_i\} \cup \{x \mapsto -1\}$, and let $\mathbb{D}_i$ be a finite nonempty set of probability measures P_i on $\mathcal{X} \times \mathcal{Y}$ such that $P_i(\mathcal{X}_i \times \mathcal{Y}) = 1$ (i.e., with marginal over $\mathcal{X}$ supported only on $\mathcal{X}_i$). Let $\mathbb{D} = \left\{\frac{1}{d}\sum_{i=1}^d P_i : \forall i \in \{1, \ldots, d\}, P_i \in \mathbb{D}_i\right\}$. Note that for any choices of $P_i \in \mathbb{D}_i$ for each $i \in \{1, \ldots, d\}$, letting $P = \frac{1}{d}\sum_{i=1}^d P_i$, we have that $\forall i \in \{1, \ldots, d\}$, $\forall x \in \mathcal{X}_i$ with $P_i(\{x\} \times \mathcal{Y}) > 0$,

$$\begin{aligned} P(\{(x,+1)\}|\{x\} \times \mathcal{Y}) &= \frac{P(\{(x,+1)\})}{P(\{x\} \times \mathcal{Y})} = \frac{\frac{1}{d}\sum_{j=1}^d P_j(\{(x,+1)\})}{\frac{1}{d}\sum_{j=1}^d P_j(\{x\} \times \mathcal{Y})} \\ &= \frac{P_i(\{(x,+1)\})}{P_i(\{x\} \times \mathcal{Y})} = P_i(\{(x,+1)\}|\{x\} \times \mathcal{Y}), \end{aligned}$$

so that the conditional distribution of Y given $X = x$ (for $(X, Y) \sim P$) is specified by the conditional of Y' given $X' = x$ for $(X', Y') \sim P_i$, for the value i with $x \in \mathcal{X}_i$. Furthermore, since any $x \in \mathcal{X}_i$ has $P(\{x\} \times \mathcal{Y}) = 0$ if and only if $P_i(\{x\} \times \mathcal{Y}) = 0$, without loss we may define $P(\{(x,+1)\}|\{x\} \times \mathcal{Y}) = P_i(\{(x,+1)\}|\{x\} \times \mathcal{Y})$ for any such x. For each $i \in \{1, \ldots, d\}$ and $\varepsilon, \delta \in (0,1)$, let $\Lambda_i(\varepsilon, \delta)$ denote the minimax label complexity under $\mathbb{D}_i$ with respect to $\mathbb{C}_i$ (i.e., the value of $\Lambda_{\mathbb{D}_i}(\varepsilon, \delta)$ when $\mathbb{C} = \mathbb{C}_i$). The value $\Lambda_{\mathbb{D}}(\varepsilon, \delta)$ remains defined as usual (i.e., with respect to the set $\mathbb{C}$ specified in (63)).

Lemma 45 *Fix any $\gamma \in (2/d, 1)$, $\varepsilon \in (0, \gamma/4)$, and $\delta \in \left(0, \frac{\gamma}{4-\gamma}\right)$. If $\min_{i \in \{1,\ldots,d\}} \Lambda_i((4/\gamma)\varepsilon, \gamma) \geq 2$, then*

$$\Lambda_{\mathbb{D}}(\varepsilon, \delta) \geq (\gamma/4)d \min_{i \in \{1,\ldots,d\}} \Lambda_i((4/\gamma)\varepsilon, \gamma).$$

Proof Fix any $n \in \mathbb{N}$ with $n < (\gamma/4)d \min_{i \in \{1,\ldots,d\}} \Lambda_i((4/\gamma)\varepsilon, \gamma)$. Denote $n' = \left\lceil \frac{n}{(\gamma/2)d} \right\rceil$, and note that $n' \leq n$ and $n' < \min_{i \in \{1,\ldots,d\}} \Lambda_i((4/\gamma)\varepsilon, \gamma)$. For each $i \in \{1, \ldots, d\}$, let $P_i \in \mathbb{D}_i$, and denote $g_i^* = \operatorname{argmin}_{g \in \mathbb{C}_i} \operatorname{er}_{P_i}(g)$ (breaking ties arbitrarily). We will later optimize over the choice of these P_i. Also let $g^* = \sum_{i=1}^d g_i^* \mathbb{1}_{\mathcal{X}_i}$, the classifier that predicts with g_i^* on each respective $\mathcal{X}_i$ set; note that, since each g_i^* classifies at most one point as $+1$, we have

$g^* \in \mathbb{C}$. Denote $P = \frac{1}{d}\sum_{i=1}^d P_i$. Let $\hat{h}_P$ denote the (random) classifier produced by $\mathcal{A}(n)$ when $\mathcal{P}_{XY} = P$. Note that if $\sum_{i=1}^d \mathbb{1}\left[\mathrm{er}_{P_i}\left(\hat{h}_P\right) - \mathrm{er}_{P_i}\left(g_i^*\right) > (4/\gamma)\varepsilon\right] > (\gamma/4)d$, then

$$\begin{aligned}
\mathrm{er}_P\left(\hat{h}_P\right) - \inf_{h\in\mathbb{C}} \mathrm{er}_P(h) &= \frac{1}{d}\sum_{i=1}^d \mathrm{er}_{P_i}\left(\hat{h}_P\right) - \inf_{h\in\mathbb{C}} \frac{1}{d}\sum_{i=1}^d \mathrm{er}_{P_i}(h) \\
&\geq \frac{1}{d}\sum_{i=1}^d \mathrm{er}_{P_i}\left(\hat{h}_P\right) - \frac{1}{d}\sum_{i=1}^d \mathrm{er}_{P_i}\left(g^*\right) = \frac{1}{d}\sum_{i=1}^d \left(\mathrm{er}_{P_i}\left(\hat{h}_P\right) - \mathrm{er}_{P_i}\left(g_i^*\right)\right) \\
&\geq \frac{1}{d}\sum_{i=1}^d \mathbb{1}\left[\mathrm{er}_{P_i}\left(\hat{h}_P\right) - \mathrm{er}_{P_i}\left(g_i^*\right) > (4/\gamma)\varepsilon\right](4/\gamma)\varepsilon > \varepsilon.
\end{aligned}$$

Therefore,

$$\begin{aligned}
&\mathbb{P}\left(\mathrm{er}_P\left(\hat{h}_P\right) - \inf_{h\in\mathbb{C}} \mathrm{er}_P(h) > \varepsilon\right) \geq \mathbb{P}\left(\sum_{i=1}^d \mathbb{1}\left[\mathrm{er}_{P_i}\left(\hat{h}_P\right) - \mathrm{er}_{P_i}\left(g_i^*\right) > (4/\gamma)\varepsilon\right] > (\gamma/4)d\right) \\
&= 1 - \mathbb{P}\left(\sum_{i=1}^d \mathbb{1}\left[\mathrm{er}_{P_i}\left(\hat{h}_P\right) - \mathrm{er}_{P_i}\left(g_i^*\right) > (4/\gamma)\varepsilon\right] \leq (\gamma/4)d\right) \\
&= 1 - \mathbb{P}\left(\sum_{i=1}^d \left(1 - \mathbb{1}\left[\mathrm{er}_{P_i}\left(\hat{h}_P\right) - \mathrm{er}_{P_i}\left(g_i^*\right) > (4/\gamma)\varepsilon\right]\right) \geq (1-\gamma/4)d\right) \\
&\geq 1 - \frac{1}{(1-\gamma/4)d}\sum_{i=1}^d \left(1 - \mathbb{P}\left(\mathrm{er}_{P_i}\left(\hat{h}_P\right) - \mathrm{er}_{P_i}\left(g_i^*\right) > (4/\gamma)\varepsilon\right)\right) \\
&= -\frac{\gamma}{4-\gamma} + \frac{4}{4-\gamma}\frac{1}{d}\sum_{i=1}^d \mathbb{P}\left(\mathrm{er}_{P_i}\left(\hat{h}_P\right) - \mathrm{er}_{P_i}\left(g_i^*\right) > (4/\gamma)\varepsilon\right), \qquad (64)
\end{aligned}$$

where the second inequality is due to Markov's inequality and linearity of expectations.

Now we note that there is a simple reduction from the problem of learning with $\mathbb{C}_i$ under P_i to the problem of learning with $\mathbb{C}$ under P. Specifically, for a given i.i.d. P_i-distributed sequence $(X_{i1}, Y_{i1}), (X_{i2}, Y_{i2}), \ldots$, we can construct i.i.d. P-distributed random variables $(X_1', Y_1'), (X_2', Y_2'), \ldots$, as follows. For each $j \in \{1, \ldots, d\} \setminus \{i\}$, let $(X_{j1}, Y_{j1}), (X_{j2}, Y_{j2}), \ldots$ be independent and P_j-distributed, and independent over j, and all independent from the (X_{it}, Y_{it}) sequence. Let $j_1, j_2, \ldots$ be independent Uniform($\{1, \ldots, d\}$) random variables (also independent from the above sequences). Then for each $t \in \mathbb{N}$, let $r_t = \sum_{s=1}^t \mathbb{1}[j_s = j_t]$, and define $(X_t', Y_t') = (X_{j_t r_t}, Y_{j_t r_t})$. One can easily verify that this these (X_t', Y_t') are independent and P-distributed. Now we can construct an active learning algorithm for the problem of learning with $\mathbb{C}_i$ under P_i, given the budget $n' \leq n$, as follows. We execute the algorithm $\mathcal{A}(n)$. If at any time it requests the label Y_t' of some X_t' in the sequence such that $j_t \neq i$, then we simply use the value $Y_t' = Y_{j_t r_t}$ (which, for the purpose of this reduction, is considered an accessible quantity). Otherwise, if $\mathcal{A}(n)$ requests the label Y_t' of some X_t' in the sequence such that $j_t = i$, then our algorithm will request the label Y_{ir_t} and provide that as the value of Y_t' to be used in the execution of $\mathcal{A}(n)$. If at any time $\mathcal{A}(n)$ has already requested n' labels Y_t' such that $j_t = i$, and attempts to request another label

Y_t' with $j_t = i$, our algorithm simply returns an arbitrary classifier, and this is considered a "failure" event. Otherwise, upon termination of $\mathcal{A}(n)$, our algorithm halts and returns the classifier $\mathcal{A}(n)$ produces. Note that this is a valid active learning algorithm for the problem of learning $\mathbb{C}_i$ under P_i with budget n', since the algorithm requests at most n' labels from the P_i-distributed sequence. In particular, in this reduction, we are thinking of the samples (X_t', Y_t') with $j_t \neq i$ as simply part of the internal randomness of the learning algorithm.

Let $\hat{h}'_{P,i}$ denote the classifier returned by the algorithm constructed via this reduction. Furthermore, if we consider also the classifier $\hat{h}_{P,i}$ returned by $\mathcal{A}(n)$ when run (unmodified) on the P-distributed sequence $(X_1', Y_1'), (X_2', Y_2'), \ldots$, and denote by $n'_{P,i}$ the number of labels Y_t' with $j_t = i$ that this unmodified $\mathcal{A}(n)$ requests, then on the event that $n'_{P,i} \leq n'$, we have $\hat{h}'_{P,i} = \hat{h}_{P,i}$. Additionally, let $n_{P,i}$ denote the number of labels Y_t requested by $\mathcal{A}(n)$ with $X_t \in \mathcal{X}_i$ (when $\mathcal{A}(n)$ is run with the sequence $\{(X_t, Y_t)\}_{t=1}^{\infty}$), and note that the sequences $\{(X_t', Y_t')\}_{t=1}^{\infty}$ and $\{(X_t, Y_t)\}_{t=1}^{\infty}$ are distributionally equivalent, so that $(\hat{h}_{P,i}, n'_{P,i})$ and $(\hat{h}_P, n_{P,i})$ are distributionally equivalent as well. Therefore,

$$
\begin{aligned}
&\mathbb{P}\left(\mathrm{er}_{P_i}\left(\hat{h}_P\right) - \mathrm{er}_{P_i}(g_i^*) > (4/\gamma)\varepsilon\right) \geq \mathbb{P}\left(\mathrm{er}_{P_i}\left(\hat{h}_P\right) - \mathrm{er}_{P_i}(g_i^*) > (4/\gamma)\varepsilon \text{ and } n_{P,i} \leq n'\right) \\
&= \mathbb{P}\left(\mathrm{er}_{P_i}\left(\hat{h}_{P,i}\right) - \mathrm{er}_{P_i}(g_i^*) > (4/\gamma)\varepsilon \text{ and } n'_{P,i} \leq n'\right) \\
&= \mathbb{P}\left(\mathrm{er}_{P_i}\left(\hat{h}'_{P,i}\right) - \mathrm{er}_{P_i}(g_i^*) > (4/\gamma)\varepsilon \text{ and } n'_{P,i} \leq n'\right) \\
&= \mathbb{P}\left(\mathrm{er}_{P_i}\left(\hat{h}'_{P,i}\right) - \mathrm{er}_{P_i}(g_i^*) > (4/\gamma)\varepsilon\right) - \mathbb{P}\left(\mathrm{er}_{P_i}\left(\hat{h}'_{P,i}\right) - \mathrm{er}_{P_i}(g_i^*) > (4/\gamma)\varepsilon \text{ and } n'_{P,i} > n'\right) \\
&\geq \mathbb{P}\left(\mathrm{er}_{P_i}\left(\hat{h}'_{P,i}\right) - \mathrm{er}_{P_i}(g_i^*) > (4/\gamma)\varepsilon\right) - \mathbb{P}\left(n'_{P,i} > n'\right) \\
&= \mathbb{P}\left(\mathrm{er}_{P_i}\left(\hat{h}'_{P,i}\right) - \mathrm{er}_{P_i}(g_i^*) > (4/\gamma)\varepsilon\right) - \mathbb{P}\left(n_{P,i} > n'\right) \\
&\geq \mathbb{P}\left(\mathrm{er}_{P_i}\left(\hat{h}'_{P,i}\right) - \mathrm{er}_{P_i}(g_i^*) > (4/\gamma)\varepsilon\right) - \frac{\mathbb{E}[n_{P,i}]}{n'},
\end{aligned}
$$

where this last inequality is due to Markov's inequality.

Applying this to every $i \in \{1, \ldots, d\}$, this implies

$$
\begin{aligned}
&\frac{1}{d}\sum_{i=1}^{d} \mathbb{P}\left(\mathrm{er}_{P_i}\left(\hat{h}_P\right) - \mathrm{er}_{P_i}\left(g_i^*\right) > (4/\gamma)\varepsilon\right) \\
&\qquad\qquad \geq -\frac{1}{dn'}\sum_{i=1}^{d}\mathbb{E}[n_{P,i}] + \frac{1}{d}\sum_{i=1}^{d}\mathbb{P}\left(\mathrm{er}_{P_i}\left(\hat{h}'_{P,i}\right) - \mathrm{er}_{P_i}\left(g_i^*\right) > (4/\gamma)\varepsilon\right).
\end{aligned}
$$

By linearity of the expectation, $\frac{1}{dn'}\sum_{i=1}^{d}\mathbb{E}[n_{P,i}] = \frac{1}{dn'}\mathbb{E}\left[\sum_{i=1}^{d} n_{P,i}\right] \leq \frac{n}{dn'} \leq \frac{\gamma}{2}$, so that the above is at least

$$
-\frac{\gamma}{2} + \frac{1}{d}\sum_{i=1}^{d}\mathbb{P}\left(\mathrm{er}_{P_i}\left(\hat{h}'_{P,i}\right) - \mathrm{er}_{P_i}\left(g_i^*\right) > (4/\gamma)\varepsilon\right).
$$

Plugging this into (64), we have that

$$
\mathbb{P}\Big(\mathrm{er}_P\Big(\hat{h}_P\Big) - \inf_{h\in\mathbb{C}}\mathrm{er}_P(h) > \varepsilon\Big) \geq -\frac{3\gamma}{4-\gamma} + \frac{4}{4-\gamma}\frac{1}{d}\sum_{i=1}^{d}\mathbb{P}\Big(\mathrm{er}_{P_i}\Big(\hat{h}'_{P,i}\Big) - \mathrm{er}_{P_i}(g_i^*) > (4/\gamma)\varepsilon\Big).
$$

The above strategy, producing $\hat{h}'_{P,i}$, is a valid active learning algorithm (with budget n') for any choices of the probability measures P_j, $j \in \{1,\ldots,d\} \setminus \{i\}$. We may therefore consider its behavior if we choose these at random. Specifically, for any probability measure $\Pi^{\setminus i}$ over $\times_{j\neq i}\mathbb{D}_j$, let $\{\tilde{P}_{j,\Pi^{\setminus i}}\}_{j\neq i} \sim \Pi^{\setminus i}$, and for any $P_i \in \mathbb{D}_i$, let $\tilde{P}_{\Pi^{\setminus i},P_i} = \frac{1}{d}P_i + \frac{1}{d}\sum_{j\neq i}\tilde{P}_{j,\Pi^{\setminus i}}$. Then $\hat{h}'_{\tilde{P}_{\Pi^{\setminus i},P_i},i}$ is the output of a valid active learning algorithm (with budget n'); in particular, here we are considering the $\tilde{P}_{j,\Pi^{\setminus i}}$ as internal random variables to the algorithm (along with their corresponding (X_{jt},Y_{jt}) samples used in the algorithm, which are now considered conditionally independent given $\{\tilde{P}_{j,\Pi^{\setminus i}}\}_{j\neq i}$, where each (X_{jt},Y_{jt}) has conditional distribution $\tilde{P}_{j,\Pi^{\setminus i}}$): that is, random variables that are independent from the data sequence $(X_{i1},Y_{i1}),(X_{i2},Y_{i2}),\ldots$. Now note that, since $n' < \Lambda_i((4/\gamma)\varepsilon,\gamma)$,

$$\max_{P_i\in\mathbb{D}_i}\mathbb{P}\left(\mathrm{er}_{P_i}\left(\hat{h}'_{\tilde{P}_{\Pi^{\setminus i},P_i},i}\right) - \inf_{g\in\mathbb{C}_i}\mathrm{er}_{P_i}(g) > (4/\gamma)\varepsilon\right) > \gamma. \tag{65}$$

For any given sequence $P_1,\ldots,P_d$, with $P_j \in \mathbb{D}_i$ for each $j \in \{1,\ldots,d\}$, for every $i \in \{1,\ldots,d\}$, denote $\psi_i(P_i,\{P_j\}_{j\neq i}) = \mathbb{P}\left(\mathrm{er}_{P_i}\left(\hat{h}'_{P,i}\right) - \inf_{g\in\mathbb{C}_i}\mathrm{er}_{P_i}(g) > (4/\gamma)\varepsilon\right)$, where $P = \frac{1}{d}\sum_{j=1}^d P_j$ as above. Then, by the law of total probability, (65) may be restated as

$$\max_{P_i\in\mathbb{D}_i}\mathbb{E}\left[\psi_i\left(P_i,\left\{\tilde{P}_{j,\Pi^{\setminus i}}\right\}_{j\neq i}\right)\right] > \gamma.$$

Since this holds for every choice of $\Pi^{\setminus i}$, we have that

$$\inf_{\Pi^{\setminus i}}\max_{P_i\in\mathbb{D}_i}\mathbb{E}\left[\psi_i\left(P_i,\left\{\tilde{P}_{j,\Pi^{\setminus i}}\right\}_{j\neq i}\right)\right] \geq \gamma.$$

Since each $\mathbb{D}_j$ is finite, by the minimax theorem (von Neumann, 1928; von Neumann and Morgenstern, 1944), for each $i \in \{1,\ldots,d\}$, there exists a probability measure Π_i over $\mathbb{D}_i$ such that, if $\tilde{P}_i \sim \Pi_i$ (independent from every $\{\tilde{P}_{j,\Pi^{\setminus i}}\}_{j\neq i}$), then

$$\inf_{\Pi^{\setminus i}}\mathbb{E}\left[\psi_i\left(\tilde{P}_i,\left\{\tilde{P}_{j,\Pi^{\setminus i}}\right\}_{j\neq i}\right)\right] = \inf_{\Pi^{\setminus i}}\max_{P_i\in\mathbb{D}_i}\mathbb{E}\left[\psi_i\left(P_i,\left\{\tilde{P}_{j,\Pi^{\setminus i}}\right\}_{j\neq i}\right)\right].$$

In particular, taking these $\{\tilde{P}_i\}_{i=1}^d$ to be independent, we have that $\forall i \in \{1,\ldots,d\}$,

$$\mathbb{E}\left[\psi_i\left(\tilde{P}_i,\left\{\tilde{P}_j\right\}_{j\neq i}\right)\right] \geq \inf_{\Pi^{\setminus i}}\mathbb{E}\left[\psi_i\left(\tilde{P}_i,\left\{\tilde{P}_{j,\Pi^{\setminus i}}\right\}_{j\neq i}\right)\right] = \inf_{\Pi^{\setminus i}}\max_{P_i\in\mathbb{D}_i}\mathbb{E}\left[\psi_i\left(P_i,\left\{\tilde{P}_{j,\Pi^{\setminus i}}\right\}_{j\neq i}\right)\right] \geq \gamma.$$

Thus,

$$\sup_{\substack{P_i\in\mathbb{D}_i:\\ i\in\{1,\ldots,d\}}}\sum_{i=1}^d\psi_i(P_i,\{P_j\}_{j\neq i}) \geq \mathbb{E}\left[\sum_{i=1}^d\psi_i\left(\tilde{P}_i,\left\{\tilde{P}_j\right\}_{j\neq i}\right)\right] = \sum_{i=1}^d\mathbb{E}\left[\psi_i\left(\tilde{P}_i,\left\{\tilde{P}_j\right\}_{j\neq i}\right)\right] \geq \gamma d.$$

Altogether, we have that

$$\begin{aligned}\sup_{\substack{P_i\in\mathbb{D}_i:\\ i\in\{1,\ldots,d\}}}\mathbb{P}\left(\mathrm{er}_P\left(\hat{h}_P\right) - \inf_{h\in\mathbb{C}}\mathrm{er}_P(h) > \varepsilon\right) &\geq -\frac{3\gamma}{4-\gamma} + \frac{4}{4-\gamma}\frac{1}{d}\sup_{\substack{P_i\in\mathbb{D}_i:\\ i\in\{1,\ldots,d\}}}\sum_{i=1}^d\psi_i\left(P_i,\{P_j\}_{j\neq i}\right)\\ &\geq -\frac{3\gamma}{4-\gamma} + \frac{4\gamma}{4-\gamma} = \frac{\gamma}{4-\gamma} > \delta.\end{aligned}$$

Since this holds for any active learning algorithm $\mathcal{A}$ and $n < (\gamma/4)d \min_{i\in\{1,\ldots,d\}} \Lambda_i((4/\gamma)\varepsilon, \gamma)$, the lemma follows. ■

With this lemma in hand, we can now plug in various sets $\mathbb{D}_i$ to obtain lower bounds for learning with this set $\mathbb{C}$ under various noise models. In particular, we can make use of the constructions of lower bounds on $\Lambda_i(\varepsilon, \delta)$ given in the proofs of the theorems in Section 5, noting that the VC dimension of $\mathbb{C}_i$ is 1, and the star number of $\mathbb{C}_i$ is $\lfloor \mathfrak{s}/d \rfloor$. Note that, in the case $d \lesssim 1$, the lower bounds in each of these theorems already match their respective upper bounds up to constant and logarithmic factors (using the lower bound from Theorem 3 as a lower bound on $\Lambda_{\mathrm{BN}(\beta)}(\varepsilon, \delta)$ for β near 0). We may therefore suppose $d \geq 32$ for the remainder of this subsection.

D.1.1 The Realizable Case

For the realizable case, for each $i \in \{1, \ldots, d\}$ and $t \in \{1, \ldots, \lfloor \mathfrak{s}/d \rfloor\}$, let $\mathcal{P}_{it}$ be a uniform distribution on $\{\lfloor \mathfrak{s}/d \rfloor (i-1) + 1, \ldots, \lfloor \mathfrak{s}/d \rfloor (i-1) + t\} \subseteq \mathcal{X}_i$, and let $\mathbb{D}_i$ denote the set of probability measures P_i in RE having marginal over $\mathcal{X}$ among $\{\mathcal{P}_{it} : 1 \leq t \leq \lfloor \mathfrak{s}/d \rfloor\}$ and having $f^\star_{P_i} \in \mathbb{C}_i$. Noting that the star number of $\mathbb{C}_i$ is $\lfloor \mathfrak{s}/d \rfloor$ and that $\mathcal{X}_i$ is a (maximal) star set for $\mathbb{C}_i$, and recalling that the first term in the "max" in the lower bound of Theorem 3 was proven in Appendix B.1 under the uniform marginal distribution on the first t elements of a maximal star set (for an appropriate value of t, of size at least 1 and at most the star number), we have that for $\varepsilon \in \left(0, \frac{1}{9 \cdot 16}\right)$,

$$\Lambda_i(16\varepsilon, 1/4) \gtrsim \min\left\{\frac{\mathfrak{s}}{d}, \frac{1}{\varepsilon}\right\}.$$

Therefore, Lemma 45 (with $\gamma = 1/4$) implies that for $\mathbb{D} = \left\{\frac{1}{d}\sum_{i=1}^d P_i : \forall i \in \{1, \ldots, d\}, P_i \in \mathbb{D}_i\right\}$, $\forall \delta \in \left(0, \frac{1}{15}\right)$,

$$\Lambda_{\mathbb{D}}(\varepsilon, \delta) \gtrsim \min\left\{\mathfrak{s}, \frac{d}{\varepsilon}\right\}.$$

Furthermore, for each choice of $P_1, \ldots, P_d$ (with each $P_i \in \mathbb{D}_i$), by construction, every $i \in \{1, \ldots, d\}$ has at most one $x \in \mathcal{X}_i$ with $P_i(\{(x, +1)\}|\{x\} \times \mathcal{Y}) = 1$, and every other x' in $\mathcal{X}_i$ has $P_i(\{(x', +1)\}|\{x'\} \times \mathcal{Y}) = 0$. Therefore, since $P(\{(x, +1)\}|\{x\} \times \mathcal{Y}) = P_i(\{(x, +1)\}|\{x\} \times \mathcal{Y})$ for every $x \in \mathcal{X}_i$, for $P = \frac{1}{d}\sum_{j=1}^d P_j$, we have that there are at most d points x in $\bigcup_{i=1}^d \mathcal{X}_i$ with $P(\{(x, +1)\}|\{x\} \times \mathcal{Y}) = 1$, and all other points x in $\bigcup_{i=1}^d \mathcal{X}_i$ have $P(\{(x, +1)\}|\{x\} \times \mathcal{Y}) = 0$. In particular, this implies that for $(X, Y) \sim P$, $\mathbb{P}(f^\star_P(X) \neq Y | X \in \bigcup_{i=1}^d \mathcal{X}_i) = 0$. Since we also have that $\forall t \in \mathbb{N} \setminus \bigcup_{i=1}^d \mathcal{X}_i$, $P(\{t\} \times \mathcal{Y}) = 0$, we can take $f^\star_P(x) = -1$ for every $x \in \mathcal{X} \setminus \bigcup_{i=1}^d \mathcal{X}_i$ while guaranteeing $\mathrm{er}_P(f^\star_P) = 0$. Since $\bigcup_{i=1}^d \mathcal{X}_i \subseteq \{1, \ldots, \mathfrak{s}\}$, we also have that $f^\star_P \in \mathbb{C}$. Together, these facts imply $P \in \mathrm{RE}$. Thus, $\mathbb{D} \subseteq \mathrm{RE}$, which implies $\Lambda_{\mathrm{RE}}(\varepsilon, \delta) \geq \Lambda_{\mathbb{D}}(\varepsilon, \delta)$, so that

$$\Lambda_{\mathrm{RE}}(\varepsilon, \delta) \gtrsim \min\left\{\mathfrak{s}, \frac{d}{\varepsilon}\right\}$$

as well. Since the upper bound in Theorem 3 is within a factor proportional to $\mathrm{Log}(1/\varepsilon)$ of this,[16] this establishes that the upper bound is sometimes tight to within a factor proportional to $\mathrm{Log}(1/\varepsilon)$.

D.1.2 Bounded Noise

In the case of bounded noise, fix any $\beta \in (0, 1/2)$ and any $\varepsilon \in (0, (1-2\beta)/(256e))$. Take $\zeta = \frac{32e\varepsilon}{1-2\beta}$ and $k = \min\{\lfloor \mathfrak{s}/d \rfloor - 1, \lfloor 1/\zeta \rfloor\}$, and for each $i \in \{1, \ldots, d\}$, let $\mathbb{D}_i$ be defined as the set $\mathrm{RR}(k, \zeta, \beta)$ in Lemma 26, as applied to the hypothesis class $\mathbb{C}_i$ with $\{x_1, \ldots, x_{k+1}\} = \{\lfloor \mathfrak{s}/d \rfloor (i-1) + 1, \ldots, \lfloor \mathfrak{s}/d \rfloor (i-1) + k + 1\}$, $h_0 = -1$, and $h_j = 2\mathbb{1}_{\{\lfloor \mathfrak{s}/d \rfloor (i-1)+j\}} - 1$ for each $j \in \{1, \ldots, k\}$. Then Lemma 26 implies

$$\Lambda_i(16e\varepsilon, 1/(4e)) \geq \frac{\beta(k-1)}{3(1-2\beta)^2} \gtrsim \frac{\beta}{(1-2\beta)^2} \min\left\{\frac{\mathfrak{s}}{d}, \frac{1-2\beta}{\varepsilon}\right\}.$$

Furthermore, recall from the definition of $\mathrm{RR}(k, \zeta, \beta)$ in Section A.2 that $\mathbb{D}_i$ is a finite set of probability measures, and every $P_i \in \mathbb{D}_i$ has $P_i((\mathcal{X} \setminus \{x_1, \ldots, x_{k+1}\}) \times \mathcal{Y}) = 0$. In particular, note that $\{x_1, \ldots, x_{k+1}\} \subseteq \mathcal{X}_i$ in this case. Furthermore, every $P_i \in \mathbb{D}_i$ has $\forall x \in \{x_1, \ldots, x_k\}$, $P_i(\{(x, +1)\}|\{x\} \times \mathcal{Y}) \in \{\beta, 1-\beta\}$, and at most one $x \in \{x_1, \ldots, x_k\}$ has $P_i(\{(x, +1)\}|\{x\} \times \mathcal{Y}) = 1 - \beta$, while $P_i(\{(x_{k+1}, +1)\}|\{x_{k+1}\} \times \mathcal{Y}) = 0$. Thus, for any choices of $P_i \in \mathbb{D}_i$ for each $i \in \{1, \ldots, d\}$, the probability measure $P = \frac{1}{d}\sum_{i=1}^{d} P_i$ satisfies the property that, $\forall x \in \mathcal{X}$ with $P(\{x\} \times \mathcal{Y}) > 0$, $P(\{(x, +1)\}|\{x\} \times \mathcal{Y}) \in \{0, \beta, 1-\beta\}$, and there are at most d values $x \in \mathcal{X}$ with $P(\{x\} \times \mathcal{Y}) > 0$ and $P(\{(x, +1)\}|\{x\} \times \mathcal{Y}) = 1 - \beta$. In particular, this implies that without loss, we can take $f_P^\star \in \mathbb{C}$, and furthermore that $P \in \mathrm{BN}(\beta)$. Thus, for the set $\mathbb{D} = \left\{\frac{1}{d}\sum_{i=1}^{d} P_i : \forall i \in \{1, \ldots, d\}, P_i \in \mathbb{D}_i\right\}$, we have $\mathbb{D} \subseteq \mathrm{BN}(\beta)$. Lemma 45 (with $\gamma = 1/(4e)$) then implies that $\forall \delta \in \left(0, \frac{1}{16e-1}\right)$,

$$\Lambda_{\mathrm{BN}(\beta)}(\varepsilon, \delta) \geq \Lambda_{\mathbb{D}}(\varepsilon, \delta) \gtrsim d \min_{i \in \{1, \ldots, d\}} \Lambda_i(16e\varepsilon, 1/(4e)) \gtrsim \frac{\beta}{(1-2\beta)^2} \min\left\{\mathfrak{s}, \frac{(1-2\beta)d}{\varepsilon}\right\}.$$

For β bounded away from 0, the upper bound in Theorem 4 is within a polylog $\left(\frac{d}{\varepsilon\delta}\right)$ factor of this, so that this establishes that the upper bound is sometimes tight to within logarithmic factors when β is bounded away from 0. Furthermore, since $\mathrm{RE} \subseteq \mathrm{BN}(\beta)$, the above result for sometimes-tightness of the upper bound in the realizable case implies that the upper bound in Theorem 4 is also sometimes tight to within logarithmic factors for any β near 0.

D.1.3 Tsybakov Noise

For the case of Tsybakov noise, the tightness (up to logarithmic factors) of the upper bound for $\alpha \leq 1/2$ is already established by the lower bound for that case in Theorem 5. Thus, it remains only to consider $\alpha \in (1/2, 1)$. Fix any values $a \in [4, \infty)$, $\alpha \in (1/2, 1)$, and $\varepsilon \in \left(0, 1/(2^{11}a^{1/\alpha})\right)$, let a' be as in the definition of $\mathrm{TN}(a, \alpha)$, and let

$$k = \min\left\{\left\lfloor \frac{\mathfrak{s}}{d} \right\rfloor - 1, \left\lfloor \frac{(a')^{\frac{\alpha-1}{\alpha}}}{64\varepsilon} \right\rfloor, \left\lfloor \frac{a'}{64\varepsilon} 4^{-\frac{1}{1-\alpha}} \right\rfloor\right\},$$

16. Note that, although $\frac{\mathfrak{s}d}{\mathrm{Log}(\mathfrak{s})}$ can sometimes be much smaller than $\mathfrak{s} \wedge \frac{d}{\varepsilon}$, we always have $\mathfrak{s} \wedge \frac{d}{\varepsilon} \lesssim \frac{\mathfrak{s}d}{\mathrm{Log}(\mathfrak{s})}\mathrm{Log}\left(\frac{1}{\varepsilon}\right)$, so that this $\mathfrak{s} \wedge \frac{d}{\varepsilon}$ lower bound does not contradict the $\frac{\mathfrak{s}d}{\mathrm{Log}(\mathfrak{s})}\mathrm{Log}\left(\frac{1}{\varepsilon}\right)$ upper bound.

$\beta = \frac{1}{2} - \left(\frac{k64\varepsilon}{a'}\right)^{1-\alpha}$, and $\zeta = \frac{128\varepsilon}{1-2\beta}$. Note that $\zeta \in (0,1)$, $\beta \in [1/4, 1/2)$, and $2 \leq k \leq \min\{\lfloor \mathfrak{s}/d \rfloor - 1, \lfloor 1/\zeta \rfloor\}$ (following the arguments from the proof of Theorem 5, with ε replaced by 64ε). Furthermore, $\forall i \in \{1,\ldots,d\}$, let $\mathbb{D}_i$ be the set $\mathrm{RR}(k,\zeta,\beta)$ in Lemma 26, as applied to the class $\mathbb{C}_i$, with $\{x_1,\ldots,x_{k+1}\} = \{\lfloor \mathfrak{s}/d \rfloor (i-1)+1, \ldots, \lfloor \mathfrak{s}/d \rfloor (i-1)+k+1\}$, $h_0 = -1$, and $h_j = 2\mathbb{1}_{\{\lfloor \mathfrak{s}/d \rfloor (i-1)+j\}} - 1$ for each $j \in \{1,\ldots,k\}$. Thus, by Lemma 26,

$$\Lambda_i(64\varepsilon, 1/16) \geq \frac{\beta(k-1)\ln(4)}{3(1-2\beta)^2} \gtrsim \left(\frac{\varepsilon}{a'}\right)^{2\alpha-2} k^{2\alpha-1}$$
$$\gtrsim a^2 \left(\frac{1}{\varepsilon}\right)^{2-2\alpha} \min\left\{\frac{\mathfrak{s}}{d}, \frac{(a')^{\frac{\alpha-1}{\alpha}}}{\varepsilon}, \frac{a'}{\varepsilon} 4^{-\frac{1}{1-\alpha}}\right\}^{2\alpha-1} \gtrsim a^2 \left(\frac{1}{\varepsilon}\right)^{2-2\alpha} \min\left\{\frac{\mathfrak{s}}{d}, \frac{1}{a^{1/\alpha}\varepsilon}\right\}^{2\alpha-1},$$

where this last inequality relies on the fact (established in the proof of Theorem 5) that $(a')^{\frac{\alpha-1}{\alpha}} \leq a' 4^{-\frac{1}{1-\alpha}}$.

We note that any $P_i \in \mathbb{D}_i$ has $P_i((\mathcal{X} \setminus \{\lfloor \mathfrak{s}/d \rfloor (i-1)+1, \ldots, \lfloor \mathfrak{s}/d \rfloor (i-1)+k+1\}) \times \mathcal{Y}) = 0$. Without loss of generality, suppose each $P_i \in \mathbb{D}_i$ has $\eta(x; P_i) = 0$ for every $x \in \mathcal{X} \setminus \{\lfloor \mathfrak{s}/d \rfloor (i-1)+1, \ldots, \lfloor \mathfrak{s}/d \rfloor (i-1)+k+1\}$. As in the proof of the lower bound in Theorem 5, we note that any $P_i \in \mathbb{D}_i$ has $P_i((x,y) : |\eta(x;P_i) - 1/2| \leq t) \leq a' t^{\alpha/(1-\alpha)}$ for every $t > 0$, and furthermore that $f^\star_{P_i}(\cdot) = \mathrm{sign}(2\eta(\cdot; P_i) - 1)$, which has at most one x with $f^\star_{P_i}(x_i) = +1$ (by definition of $\mathrm{RR}(k,\zeta,\beta)$ in Section A.2). This further implies that, for any choices of $P_i \in \mathbb{D}_i$ for each $i \in \{1,\ldots,d\}$, the probability measure $P = \frac{1}{d}\sum_{i=1}^d P_i$ has support for its marginal over $\mathcal{X}$ only in $\bigcup_{i=1}^d \{\lfloor \mathfrak{s}/d \rfloor (i-1)+1, \ldots, \lfloor \mathfrak{s}/d \rfloor (i-1)+k+1\}$, and for each $i \in \{1,\ldots,d\}$, $\forall x \in \{\lfloor \mathfrak{s}/d \rfloor (i-1)+1, \ldots, \lfloor \mathfrak{s}/d \rfloor (i-1)+k+1\}$, $\eta(x;P) = \eta(x;P_i)$, while we may take $\eta(x;P) = 0$ for every $x \notin \bigcup_{i=1}^d \{\lfloor \mathfrak{s}/d \rfloor (i-1)+1, \ldots, \lfloor \mathfrak{s}/d \rfloor (i-1)+k+1\}$. Therefore, $f^\star_P$ has at most d points $x \in \bigcup_{i=1}^d \mathcal{X}_i$ with $f^\star_P(x) = +1$, and $f^\star_P(x) = -1$ for all other $x \in \mathcal{X}$: that is, $f^\star_P \in \mathbb{C}$. Additionally, since the supports of the marginals of the P_i distributions over $\mathcal{X}$ are disjoint, we have that $\forall t > 0$,

$$P\left((x,y) : |\eta(x;P) - 1/2| \leq t\right) = \frac{1}{d}\sum_{i=1}^d P_i\left((x,y) : |\eta(x;P) - 1/2| \leq t\right)$$
$$= \frac{1}{d}\sum_{i=1}^d P_i\left((x,y) : |\eta(x;P_i) - 1/2| \leq t\right) \leq \frac{1}{d}\sum_{i=1}^d a' t^{\alpha/(1-\alpha)} = a' t^{\alpha/(1-\alpha)}.$$

Thus, the set $\mathbb{D} = \left\{\frac{1}{d}\sum_{i=1}^d P_i : \forall i \in \{1,\ldots,d\}, P_i \in \mathbb{D}_i\right\}$ satisfies $\mathbb{D} \subseteq \mathrm{TN}(a,\alpha)$. Combined with the fact that each set $\mathbb{D}_i$ is finite (by the definition of $\mathrm{RR}(k,\zeta,\beta)$ in Section A.2), Lemma 45 (with $\gamma = 1/16$) implies that $\forall \delta \in \left(0, \frac{1}{63}\right)$,

$$\Lambda_{\mathrm{TN}(a,\alpha)}(\varepsilon,\delta) \geq \Lambda_{\mathbb{D}}(\varepsilon,\delta) \gtrsim d \min_{i \in \{1,\ldots,d\}} \Lambda_i(64\varepsilon, 1/16) \gtrsim a^2 \left(\frac{1}{\varepsilon}\right)^{2-2\alpha} \min\left\{\frac{\mathfrak{s}}{d}, \frac{1}{a^{1/\alpha}\varepsilon}\right\}^{2\alpha-1} d.$$

Since this is within logarithmic factors of the upper bound of Theorem 5, this establishes that the upper bound is sometimes tight to within logarithmic factors (for sufficiently small values of ε).

D.1.4 Benign Noise

We can establish that the upper bound in Theorem 7 is sometimes tight by reduction from the above problems. Specifically, since $\mathrm{RE} \subseteq \mathrm{BE}(\nu)$ for every $\nu \in [0, 1/2)$, for the above choice of $\mathbb{C}$ we have that $\forall \nu \in [0, 1/2]$, $\forall \varepsilon \in \left(0, \frac{1}{9 \cdot 16}\right)$, $\forall \delta \in \left(0, \frac{1}{15}\right)$,

$$\Lambda_{\mathrm{BE}(\nu)}(\varepsilon, \delta) \geq \Lambda_{\mathrm{RE}}(\varepsilon, \delta) \gtrsim \min\left\{\mathfrak{s}, \frac{d}{\varepsilon}\right\}.$$

Furthermore, the lower bound in Theorem 7 already implies that $\forall \varepsilon \in \left(0, \frac{1-2\nu}{24}\right)$, $\forall \delta \in \left(0, \frac{1}{24}\right]$,

$$\Lambda_{\mathrm{BE}(\nu)}(\varepsilon, \delta) \gtrsim \frac{\nu^2}{\varepsilon^2} d.$$

Together, we have that $\forall \nu \in [0, 1/2)$, $\forall \varepsilon \in \left(0, \frac{1-2\nu}{9 \cdot 16}\right)$, $\forall \delta \in \left(0, \frac{1}{24}\right]$,

$$\Lambda_{\mathrm{BE}(\nu)}(\varepsilon, \delta) \gtrsim \max\left\{\frac{\nu^2}{\varepsilon^2} d, \min\left\{\mathfrak{s}, \frac{d}{\varepsilon}\right\}\right\} \gtrsim \frac{\nu^2}{\varepsilon^2} d + \min\left\{\mathfrak{s}, \frac{d}{\varepsilon}\right\}.$$

Thus, the upper bound in Theorem 7 is sometimes tight to within logarithmic factors.

D.2 The Lower Bounds are Sometimes Tight

We now argue that the lower bounds in Theorems 3, 4, 5, and 7 are sometimes tight (up to logarithmic factors). First we have a general lemma. Let $\mathcal{X}_1 \subset \mathcal{X}$ and $\mathcal{X}_2 = \mathcal{X} \setminus \mathcal{X}_1$, and let $\mathbb{C}_1, \mathbb{C}_2$ be hypothesis classes such that $\forall i \in \{1, 2\}$, $\forall h \in \mathbb{C}_i$, $\forall x \in \mathcal{X} \setminus \mathcal{X}_i$, $h(x) = -1$. Further suppose that $\forall i \in \{1, 2\}$, the all-negative classifier $x \mapsto h_-(x) = -1$ is in $\mathbb{C}_i$. For each $i \in \{1, 2\}$ and $\gamma \in [0, 1]$, let $\mathbb{D}_i(\gamma)$ be a nonempty set of probability measures on $\mathcal{X} \times \mathcal{Y}$ such that $\forall P_i \in \mathbb{D}_i(\gamma)$, $P_i(\mathcal{X}_i \times \mathcal{Y}) = 1$; further suppose $\forall \gamma, \gamma' \in [0, 1]$ with $\gamma \leq \gamma'$, $\mathbb{D}_i(\gamma) \supseteq \mathbb{D}_i(\gamma')$. Also, for each $i \in \{1, 2\}$, $\gamma, \delta \in [0, 1]$, and $\varepsilon > 0$, let $\Lambda_{i,\gamma}(\varepsilon, \delta)$ denote the minimax label complexity under $\mathbb{D}_i(\gamma)$ with respect to $\mathbb{C}_i$ (i.e., the value of $\Lambda_{\mathbb{D}_i(\gamma)}(\varepsilon, \delta)$ when $\mathbb{C} = \mathbb{C}_i$). Let $\mathbb{D} = \{\gamma P_1 + (1-\gamma) P_2 : P_1 \in \mathbb{D}_1(\gamma), P_2 \in \mathbb{D}_2(1-\gamma), \gamma \in [0, 1]\}$.

Lemma 46 *For $\mathbb{C} = \mathbb{C}_1 \cup \mathbb{C}_2$, $\forall \varepsilon, \delta \in (0, 1)$,*

$$\Lambda_{\mathbb{D}}(\varepsilon, \delta) \leq 2 \sup_{\gamma \in [0,1]} \max\left\{\Lambda_{1,(\gamma - \varepsilon/8) \vee 0}\left(\frac{\varepsilon}{2(\gamma + \varepsilon/8)}, \frac{\delta}{3}\right), \Lambda_{2,(1-\gamma-\varepsilon/8) \vee 0}\left(\frac{\varepsilon}{2(1 - \gamma + \varepsilon/8)}, \frac{\delta}{3}\right)\right\}.$$

Proof For each $i \in \{1, 2\}$ and $\gamma \in [0, 1]$, let $\mathcal{A}_{\gamma,i}$ be an active learning algorithm such that, for any integer $n \geq \Lambda_{i,\gamma}\left(\frac{\varepsilon}{2(\gamma+\varepsilon/8)}, \frac{\delta}{3}\right)$, if $\mathcal{P}_{XY} \in \mathbb{D}_i(\gamma)$, then with probability at least $1 - \delta/3$, the classifier $\hat{h}$ produced by $\mathcal{A}_{\gamma,i}(n)$ satisfies $\mathrm{er}_{\mathcal{P}_{XY}}(\hat{h}) - \inf_{h \in \mathbb{C}_i} \mathrm{er}_{\mathcal{P}_{XY}}(h) \leq \frac{\varepsilon}{2(\gamma+\varepsilon/8)}$; such an algorithm is guaranteed to exist by the definition of $\Lambda_{i,\gamma}(\cdot, \cdot)$.

Now suppose $\mathcal{P}_{XY} \in \mathbb{D}$, so that $\mathcal{P}_{XY} = \gamma P_1 + (1-\gamma) P_2$ for some $\gamma \in [0, 1]$, $P_1 \in \mathbb{D}_1(\gamma)$, and $P_2 \in \mathbb{D}_2(1-\gamma)$. Let $(X_1, Y_1), (X_2, Y_2), \ldots$ be the data sequence, as usual (i.i.d. $\mathcal{P}_{XY}$). Consider an active learning algorithm $\mathcal{A}$ defined as follows. We first split the sequence of indices into three subsequences: $i_{0,k} = 2k - 1$ for $k \in \mathbb{N}$, $i_{1,1}, i_{1,2}, \ldots$ is the increasing subsequence of indices i such that $i/2 \in \mathbb{N}$ and $X_i \in \mathcal{X}_1$, and $i_{2,1}, i_{2,2}, \ldots$ is the remaining increasing subsequence (i.e., indices i such that $i/2 \in \mathbb{N}$ and $X_i \in \mathcal{X}_2$). Given a budget $n \in \mathbb{N}$, $\mathcal{A}(n)$

proceeds as follows. First, we let $m = \left\lceil \frac{128}{\varepsilon^2} \ln\left(\frac{12}{\delta}\right) \right\rceil$, $\gamma_1 = \max\left\{ \frac{1}{m}\sum_{k=1}^{m} \mathbb{1}_{\mathcal{X}_1}(X_{i_{0,k}}) - \frac{\varepsilon}{16}, 0 \right\}$, and $\gamma_2 = \max\left\{ \frac{1}{m}\sum_{k=1}^{m} \mathbb{1}_{\mathcal{X}_2}(X_{i_{0,k}}) - \frac{\varepsilon}{16}, 0 \right\}$. By Hoeffding's inequality and a union bound, with probability at least $1-\delta/3$, $\forall i \in \{1,2\}$,

$$\mathcal{P}_{XY}(\mathcal{X}_i \times \mathcal{Y}) - \frac{\varepsilon}{8} \le \gamma_i \le \mathcal{P}_{XY}(\mathcal{X}_i \times \mathcal{Y}). \tag{66}$$

Denote by H this event.

Next, for each $j \in \{1,2\}$, if the subsequence $i_{j,1}, i_{j,2}, \ldots$ is infinite, then run $\mathcal{A}_{\gamma_j,j}(\lfloor n/2 \rfloor)$ with the data subsequence $\{X_k^{(j)}\}_{k=1}^{\infty} = \{X_{i_{j,k}}\}_{k=1}^{\infty}$; if the algorithm $\mathcal{A}_{\gamma_j,j}$ requests the label for an index k (i.e., corresponding to $X_k^{(j)}$), then $\mathcal{A}(n)$ requests the corresponding label $Y_{i_{j,k}}$ and provides this value to $\mathcal{A}_{\gamma_j,j}$ as the label of $X_k^{(j)}$. Let $\hat{h}_j$ denote the classifier returned by this execution of $\mathcal{A}_{\gamma_j,j}(\lfloor n/2 \rfloor)$. On the other hand, if the subsequence $i_{j,1}, i_{j,2}, \ldots$ is finite (or empty), then we let $\hat{h}_j$ denote an arbitrary classifier. Finally, let $\mathcal{A}(n)$ return the classifier $\hat{h} = \hat{h}_1 \mathbb{1}_{\mathcal{X}_1} + \hat{h}_2 \mathbb{1}_{\mathcal{X}_2}$. In particular, note that this method requests at most n labels, since all labels are requested by one of the $\mathcal{A}_{\gamma_j,j}$ algorithms, each of which requests at most $\lfloor n/2 \rfloor$ labels.

For this method, we have that

$$\begin{aligned}
\mathrm{er}_{\mathcal{P}_{XY}}(\hat{h}) - \inf_{h\in\mathbb{C}} \mathrm{er}_{\mathcal{P}_{XY}}(h) &= \gamma \mathrm{er}_{P_1}(\hat{h}_1) + (1-\gamma)\mathrm{er}_{P_2}(\hat{h}_2) - \inf_{h\in\mathbb{C}} \left(\gamma \mathrm{er}_{P_1}(h) + (1-\gamma)\mathrm{er}_{P_2}(h)\right) \\
&\le \gamma \left(\mathrm{er}_{P_1}(\hat{h}_1) - \inf_{h\in\mathbb{C}} \mathrm{er}_{P_1}(h) \right) + (1-\gamma) \left(\mathrm{er}_{P_2}(\hat{h}_2) - \inf_{h\in\mathbb{C}} \mathrm{er}_{P_2}(h) \right).
\end{aligned}$$

For each $j \in \{1,2\}$, since every $h \in \mathbb{C} \setminus \mathbb{C}_j$ has $h(x) = h_-(x)$ for every $x \in \mathcal{X}_j$, and $h_- \in \mathbb{C}_j$, we have that $\inf_{h\in\mathbb{C}} \mathrm{er}_{P_j}(h) = \inf_{h\in\mathbb{C}_j} \mathrm{er}_{P_j}(h)$. Thus, the above implies

$$\mathrm{er}_{\mathcal{P}_{XY}}(\hat{h}) - \inf_{h\in\mathbb{C}} \mathrm{er}_{\mathcal{P}_{XY}}(h) \le \gamma \left(\mathrm{er}_{P_1}(\hat{h}_1) - \inf_{h\in\mathbb{C}_1} \mathrm{er}_{P_1}(h) \right) + (1-\gamma) \left(\mathrm{er}_{P_2}(\hat{h}_2) - \inf_{h\in\mathbb{C}_2} \mathrm{er}_{P_2}(h) \right). \tag{67}$$

If $\gamma = 0$, then with probability one, every $X_i \in \mathcal{X}_2$, and $\{(X_{i_{2,k}}, Y_{i_{2,k}})\}_{k=1}^{\infty}$ is an infinite i.i.d. P_2-distributed sequence. Furthermore, $1 - \varepsilon/8 < \gamma_2 = 1 - \varepsilon/16 < 1$, so that $\mathcal{P}_{XY} \in \mathbb{D}_2(\gamma_2)$. Thus, if $n \ge 2\Lambda_{2,1-\varepsilon/8}\left(\frac{\varepsilon}{2(1+\varepsilon/8)}, \frac{\delta}{3}\right)$, then we also have $n \ge \Lambda_{2,\gamma_2}\left(\frac{\varepsilon}{2(\gamma_2+\varepsilon/8)}, \frac{\delta}{3}\right)$ (by monotonicity of $\mathbb{D}_2(\cdot)$ and the label complexity), so that with probability at least $1-\delta/3$, $\mathrm{er}_{P_2}(\hat{h}_2) - \inf_{h\in\mathbb{C}_2} \mathrm{er}_{P_2}(h) \le \frac{\varepsilon}{2(\gamma_2+\varepsilon/8)} = \frac{\varepsilon}{2(1+\varepsilon/16)} < \frac{\varepsilon}{2}$ (here we are evaluating the label complexity guarantee of $\mathcal{A}_{\gamma_2,2}$ under the conditional distribution given γ_2, and then invoking the law of total probability and intersecting with the above probability-one event). Combined with (67), this implies $\mathrm{er}_{\mathcal{P}_{XY}}(\hat{h}) - \inf_{h\in\mathbb{C}} \mathrm{er}_{\mathcal{P}_{XY}}(h) < \frac{\varepsilon}{2}$. If $\gamma = 1$, then a symmetric argument implies that if $n \ge 2\Lambda_{1,1-\varepsilon/8}\left(\frac{\varepsilon}{2(1+\varepsilon/8)}, \frac{\delta}{3}\right)$, then with probability at least $1-\delta/3$, $\mathrm{er}_{\mathcal{P}_{XY}}(\hat{h}) - \inf_{h\in\mathbb{C}} \mathrm{er}_{\mathcal{P}_{XY}}(h) < \frac{\varepsilon}{2}$.

Otherwise, suppose $0 < \gamma < 1$. Note that, on the event H, $\gamma - \varepsilon/8 \le \gamma_1 \le \gamma$ and $1-\gamma-\varepsilon/8 \le \gamma_2 \le 1-\gamma$, so that $\mathbb{D}_1(\gamma_1) \subseteq \mathbb{D}_1((\gamma-\varepsilon/8)\vee 0)$ and $\mathbb{D}_2(\gamma_2) \subseteq \mathbb{D}_2((1-\gamma-\varepsilon/8)\vee 0)$, and hence that

$$\Lambda_{1,\gamma_1}\left(\frac{\varepsilon}{2(\gamma_1+\varepsilon/8)}, \frac{\delta}{3}\right) \le \Lambda_{1,(\gamma-\varepsilon/8)\vee 0}\left(\frac{\varepsilon}{2(\gamma+\varepsilon/8)}, \frac{\delta}{3}\right)$$

and

$$\Lambda_{2,\gamma_2}\left(\frac{\varepsilon}{2(\gamma_2+\varepsilon/8)},\frac{\delta}{3}\right) \leq \Lambda_{2,(1-\gamma-\varepsilon/8)\vee 0}\left(\frac{\varepsilon}{2(1-\gamma+\varepsilon/8)},\frac{\delta}{3}\right).$$

In this case, by the strong law of large numbers, with probability one, $\forall j \in \{1,2\}$, the sequence $i_{j,1}, i_{j,2}, \ldots$ exists and is infinite. Since the support of the marginal of P_j over $\mathcal{X}$ is contained within $\mathcal{X}_j$, and $\mathcal{X}_1$ and $\mathcal{X}_2$ are disjoint, we may observe that $(X_{i_{j,1}}, Y_{i_{j,1}})$, $(X_{i_{j,2}}, Y_{i_{j,2}}), \ldots$ are independent P_j-distributed random variables. In particular, if

$$n \geq 2\max\left\{\Lambda_{1,(\gamma-\varepsilon/8)\vee 0}\left(\frac{\varepsilon}{2(\gamma+\varepsilon/8)},\frac{\delta}{3}\right), \Lambda_{2,(1-\gamma-\varepsilon/8)\vee 0}\left(\frac{\varepsilon}{2(1-\gamma+\varepsilon/8)},\frac{\delta}{3}\right)\right\},$$

then (by the label complexity guarantee of $\mathcal{A}_{\gamma_j,j}$ applied under the conditional distribution given γ_j, combined with the law of total probability, and intersecting with the above probability-one event) there are events H_1 and H_2, each of probability at least $1-\delta/3$, such that on the event $H \cap H_1$, $\mathrm{er}_{P_1}(\hat{h}_1) - \inf_{h\in\mathbb{C}_1}\mathrm{er}_{P_1}(h) \leq \frac{\varepsilon}{2(\gamma_1+\varepsilon/8)} \leq \frac{\varepsilon}{2\gamma}$, and on the event $H \cap H_2$, $\mathrm{er}_{P_2}(\hat{h}_2) - \inf_{h\in\mathbb{C}_2}\mathrm{er}_{P_2}(h) \leq \frac{\varepsilon}{2(\gamma_2+\varepsilon/8)} \leq \frac{\varepsilon}{2(1-\gamma)}$. Therefore, on the event $H \cap H_1 \cap H_2$, the right hand side of (67) is at most $\gamma\frac{\varepsilon}{2\gamma} + (1-\gamma)\frac{\varepsilon}{2(1-\gamma)} = \varepsilon$, so that $\mathrm{er}_{\mathcal{P}_{XY}}(\hat{h}) - \inf_{h\in\mathbb{C}}\mathrm{er}_{\mathcal{P}_{XY}}(h) \leq \varepsilon$. By a union bound, the probability of $H \cap H_1 \cap H_2$ is at least $1-\delta$. Since this holds for any $\mathcal{P}_{XY} \in \mathbb{D}$, the result follows. ■

We can now apply this result with various choices of the sets $\mathbb{D}_1(\gamma)$ and $\mathbb{D}_2(\gamma)$ to obtain upper bounds for the above space $\mathbb{C}$, matching the lower bounds proven above for various noise models. Specifically, consider $\mathcal{X} = \mathbb{N}$, $\mathcal{X}_1 = \{1,\ldots,d\}$, $\mathcal{X}_2 = \{d+1, d+2, \ldots\}$, $\mathbb{C}_1 = \{x \mapsto 2\mathbb{1}_S(x) - 1 : S \subseteq \{1,\ldots,d\}\}$, and $\mathbb{C}_2 = \left\{x \mapsto 2\mathbb{1}_{\{t\}}(x) - 1 : t \in \{d+1, d+2, \ldots, \mathfrak{s}\}\right\} \cup \{x \mapsto -1\}$. Note that $\mathbb{C}_1$ and $\mathbb{C}_2$ satisfy the requirements specified above, and also that the VC dimension of $\mathbb{C}_1$ is d and the star number of $\mathbb{C}_1$ is d, while the VC dimension of $\mathbb{C}_2$ is 1 and the star number of $\mathbb{C}_2$ is $\mathfrak{s}-d$. Furthermore, take $\mathbb{C} = \{x \mapsto 2\mathbb{1}_S(x) - 1 : S \in 2^{\{1,\ldots,d\}} \cup \{\{i\} : d+1 \leq i \leq \mathfrak{s}\}\}$, and note that this satisfies $\mathbb{C} = \mathbb{C}_1 \cup \mathbb{C}_2$, and $\mathbb{C}$ has VC dimension d and star number $\mathfrak{s}$.

D.2.1 The Realizable Case

For the realizable case, we can in fact show that that lower bound in Theorem 3 is sometimes tight up to *universal constant* factors. Specifically, let $\mathbb{D}_i$ denote the set of all $P_i \in \mathrm{RE}$ with $P_i(\mathcal{X}_i \times \mathcal{Y}) = 1$, for each $i \in \{1,2\}$. For every $\gamma \in [0,1]$ and $i \in \{1,2\}$, define $\mathbb{D}_i(\gamma) = \mathbb{D}_i$. In particular, note that for any $P \in \mathrm{RE}$, for any measurable $A \subseteq \mathcal{X} \times \mathcal{Y}$, $P(A) = P(\mathcal{X}_1 \times \mathcal{Y})P(A|\mathcal{X}_1 \times \mathcal{Y}) + P(\mathcal{X}_2 \times \mathcal{Y})P(A|\mathcal{X}_2 \times \mathcal{Y})$. Furthermore, note that any $i \in \{1,2\}$ with $P(\mathcal{X}_i \times \mathcal{Y}) > 0$ has $P(\cdot \times \mathcal{Y}|\mathcal{X}_i \times \mathcal{Y})$ supported only in $\mathcal{X}_i$, and has $P(\cdot|\mathcal{X}_i \times \mathcal{Y}) \in \mathrm{RE}$, so that $P(\cdot|\mathcal{X}_i \times \mathcal{Y}) \in \mathbb{D}_i$. Thus, $P \in \mathbb{D} = \{\gamma P_1 + (1-\gamma)P_2 : P_1 \in \mathbb{D}_1, P_2 \in \mathbb{D}_2, \gamma \in [0,1]\}$. Therefore, $\mathrm{RE} \subseteq \mathbb{D}$. Together with Lemma 46, this implies $\forall \varepsilon, \delta \in (0,1)$,

$$\begin{aligned}\Lambda_{\mathrm{RE}}(\varepsilon,\delta) \leq \Lambda_{\mathbb{D}}(\varepsilon,\delta) &\leq 2\max\left\{\Lambda_{1,0}\left(\frac{\varepsilon}{2(1+\varepsilon/8)},\frac{\delta}{3}\right), \Lambda_{2,0}\left(\frac{\varepsilon}{2(1+\varepsilon/8)},\frac{\delta}{2}\right)\right\} \\ &\leq 2\max\left\{\Lambda_{1,0}\left(\frac{\varepsilon}{3},\frac{\delta}{3}\right), \Lambda_{2,0}\left(\frac{\varepsilon}{3},\frac{\delta}{2}\right)\right\},\end{aligned}$$

for $\Lambda_{i,0}(\cdot,\cdot)$ defined as above.

Now note that, since every $P_1 \in \mathbb{D}_1$ has $P_1(\cdot \times \mathcal{Y})$ supported only in $\mathcal{X}_1$, and $P_1 \in \text{RE}$, and since $\mathbb{C}_1$ contains classifiers realizing all 2^d distinct classifications of $\mathcal{X}_1$, $\exists h_{P_1} \in \mathbb{C}_1$ with $\text{er}_{P_1}(h_{P_1}) = 0$; thus, without loss, we can take $f^\star_{P_1} = h_{P_1}$, so that P_1 is in the realizable case with respect to $\mathbb{C}_1$. In particular, since there are only d points in $\mathcal{X}_1$, if we consider the active learning algorithm that (given a budget $n \geq d$) simply requests Y_i for exactly one i s.t. $X_i = x$, for each $x \in \mathcal{X}_1$ for which $\exists X_i = x$, and then returns any classifier $\hat{h}$ consistent with these labels, if $\mathcal{P}_{XY} \in \mathbb{D}_1$, with probability one every $x \in \mathcal{X}_1$ with $\mathcal{P}_{XY}(\{x\} \times \mathcal{Y}) > 0$ has some $X_i = x$, so that $\text{er}_{\mathcal{P}_{XY}}(\hat{h}) = 0$. Noting that this algorithm requests at most d labels, we have that $\forall \varepsilon, \delta \in (0,1)$,

$$\Lambda_{1,0}\left(\frac{\varepsilon}{3}, \frac{\delta}{3}\right) \leq d.$$

Similarly, since every $P_2 \in \mathbb{D}_2$ has $P_2(\cdot \times \mathcal{Y})$ supported only in $\mathcal{X}_2$, and $P_2 \in \text{RE}$, $f^\star_{P_2}$ is either equal -1 with P_2-probability one, or else $\exists x \in \{d+1, \ldots, \mathfrak{s}\}$ with $f^\star_{P_2}(x) = +1$; in either case, $\exists h_{P_2} \in \mathbb{C}_2$ with $\text{er}_{P_2}(h_{P_2}) = 0$; thus, without loss, we can take $f^\star_{P_2} = h_{P_2}$, so that P_2 is in the realizable case with respect to $\mathbb{C}_2$. Now consider an active learning algorithm that first calculates the empirical frequency $\hat{\mathcal{P}}(\{x\}) = \frac{1}{m}\sum_{i=1}^m \mathbb{1}[X_i = x]$ for each $x \in \{d+1, \ldots, \mathfrak{s}\}$ among the first $m = \left\lceil \frac{3^4}{2\varepsilon^4} \ln\left(\frac{3(\mathfrak{s}-d)}{\delta}\right)\right\rceil$ unlabeled data points. Then, for each $x \in \{d+1, \ldots, \mathfrak{s}\}$, if $\hat{\mathcal{P}}(\{x\}) > (1-\varepsilon/3)\varepsilon/3$, the algorithm requests Y_i for the first $i \in \mathbb{N}$ with $X_i = x$ (supposing the budget n has not yet been reached). If any requested value Y_i equals $+1$, then for the $x \in \{d+1, \ldots, \mathfrak{s}\}$ with $X_i = x$, the algorithm returns the classifier $x' \mapsto 2\mathbb{1}_{\{x\}}(x') - 1$. Otherwise, the algorithm returns the all-negative classifier: $x' \mapsto -1$. Denote by $\hat{h}$ the classifier returned by the algorithm. By Hoeffding's inequality and a union bound, with probability at least $1 - \delta/3$, every $x \in \{d+1, \ldots, \mathfrak{s}\}$ has $\hat{\mathcal{P}}(\{x\}) \geq \mathcal{P}_{XY}(\{x\} \times \mathcal{Y}) - (\varepsilon/3)^2$. Also, if $\mathcal{P}_{XY} \in \text{RE}$, then with probability one, every $Y_i = f^\star_{\mathcal{P}_{XY}}(X_i)$. Therefore, if $\mathcal{P}_{XY} \in \mathbb{D}_2$, on these events, every $x \in \{d+1, \ldots, \mathfrak{s}\}$ with $\mathcal{P}_{XY}(\{x\} \times \mathcal{Y}) > \varepsilon/3$ will have a label Y_i with $X_i = x$ requested by the algorithm (supposing sufficiently large n), which implies $\hat{h}(x) = f^\star_{\mathcal{P}_{XY}}(x)$. Since $f^\star_{\mathcal{P}_{XY}}$ has at most one $x \in \mathcal{X}_2$ with $f^\star_{\mathcal{P}_{XY}}(x) = +1$, and if such an x exists it must be in $\{d+1, \ldots, \mathfrak{s}\}$, if any requested $Y_i = +1$, we have $\text{er}_{\mathcal{P}_{XY}}(\hat{h}) = 0$, and otherwise either no $x \in \mathcal{X}_2$ has $f^\star_{\mathcal{P}_{XY}}(x) = +1$ or else the one such x has $\mathcal{P}_{XY}(\{x\} \times \mathcal{Y}) \leq \varepsilon/3$; in either case, we have $\text{er}_{\mathcal{P}_{XY}}(\hat{h}) = \mathcal{P}_{XY}(\{x : f^\star_{\mathcal{P}_{XY}}(x) = +1\} \times \mathcal{Y}) \leq \varepsilon/3$. Thus, regardless of whether the algorithm requests a Y_i with value $+1$, we have $\text{er}_{\mathcal{P}_{XY}}(\hat{h}) \leq \varepsilon/3$. By a union bound for the two events, we have that $\mathbb{P}(\text{er}_{\mathcal{P}_{XY}}(\hat{h}) > \varepsilon/3) \leq \delta/3$ (given a sufficiently large n). Furthermore, there are at most $\min\left\{\mathfrak{s}-d, \frac{1}{(1-\varepsilon/3)\varepsilon/3}\right\}$ points $x \in \{d+1, \ldots, \mathfrak{s}\}$ with $\hat{\mathcal{P}}(\{x\}) > (1-\varepsilon/3)\varepsilon/3$, and therefore at most this many labels Y_i are requested by the algorithm. Thus, a budget n of at least this size suffices for this guarantee. Since this holds for every $\mathcal{P}_{XY} \in \mathbb{D}_2$, we have that

$$\Lambda_{2,0}\left(\frac{\varepsilon}{3}, \frac{\delta}{3}\right) \leq \min\left\{\mathfrak{s}-d, \frac{1}{(1-\varepsilon/3)\varepsilon/3}\right\} \lesssim \min\left\{\mathfrak{s}, \frac{1}{\varepsilon}\right\}.$$

Altogether, we have that $\forall \varepsilon, \delta \in (0,1)$,

$$\Lambda_{\mathrm{RE}}(\varepsilon, \delta) \lesssim \max\left\{\min\left\{\mathfrak{s}, \frac{1}{\varepsilon}\right\}, d\right\}.$$

Thus, the lower bound in Theorem 3 is tight up to universal constant factors in this case.[17]

D.2.2 Bounded Noise

To prove that the lower bound in Theorem 4 is sometimes tight, fix any $\beta \in (0, 1/2)$, and let $\mathbb{D}_i$ denote the set of all $P_i \in \mathrm{BN}(\beta)$ with $P_i(\mathcal{X}_i \times \mathcal{Y}) = 1$, for each $i \in \{1,2\}$. For all $\gamma \in [0,1]$ and $i \in \{1,2\}$, define $\mathbb{D}_i(\gamma) = \mathbb{D}_i$. As above, note that for any $P \in \mathrm{BN}(\beta)$, for any measurable $A \subseteq \mathcal{X} \times \mathcal{Y}$, $P(A) = P(\mathcal{X}_1 \times \mathcal{Y})P(A|\mathcal{X}_1 \times \mathcal{Y}) + P(\mathcal{X}_2 \times \mathcal{Y})P(A|\mathcal{X}_2 \times \mathcal{Y})$. Furthermore, any $i \in \{1,2\}$ with $P(\mathcal{X}_i \times \mathcal{Y}) > 0$ has $P(\cdot \times \mathcal{Y}|\mathcal{X}_i \times \mathcal{Y})$ supported only on $\mathcal{X}_i$, and since $\eta(x; P(\cdot|\mathcal{X}_i \times \mathcal{Y})) = \eta(x; P)$ for every $x \in \mathcal{X}_i$, we have $P(\cdot|\mathcal{X}_i \times \mathcal{Y}) \in \mathrm{BN}(\beta)$, so that $P(\cdot|\mathcal{X}_i \times \mathcal{Y}) \in \mathbb{D}_i$. Thus, $P \in \mathbb{D} = \{\gamma P_1 + (1-\gamma)P_2 : P_1 \in \mathbb{D}_1, P_2 \in \mathbb{D}_2, \gamma \in [0,1]\}$. Therefore, $\mathrm{BN}(\beta) \subseteq \mathbb{D}$. Together with Lemma 46, this implies $\forall \varepsilon, \delta \in (0,1)$,

$$\Lambda_{\mathrm{BN}(\beta)}(\varepsilon, \delta) \leq \Lambda_{\mathbb{D}}(\varepsilon, \delta) \leq 2\max\left\{\Lambda_{1,0}\left(\frac{\varepsilon}{3}, \frac{\delta}{3}\right), \Lambda_{2,0}\left(\frac{\varepsilon}{3}, \frac{\delta}{3}\right)\right\},$$

for $\Lambda_{i,0}(\cdot,\cdot)$ defined as above.

Now note that, for each $i \in \{1,2\}$, since every $P_i \in \mathbb{D}_i$ has $P_i \in \mathrm{BN}(\beta)$, we have $f^\star_{P_i} \in \mathbb{C}$. Furthermore, since every $h \in \mathbb{C} \setminus \mathbb{C}_i$ has $h(x) = -1$ for every $x \in \mathcal{X}_i$, and the all-negative function $x \mapsto -1$ is contained in $\mathbb{C}_i$, and since $P_i(\mathcal{X}_i \times \mathcal{Y}) = 1$, without loss we can take $f^\star_{P_i} \in \mathbb{C}_i$ (i.e., there is a version of $f^\star_{P_i}$ contained in $\mathbb{C}_i$). Together with the condition on $\eta(\cdot; P_i)$ from the definition of $\mathrm{BN}(\beta)$, this implies each P_i satisfies the bounded noise condition (with parameter β) with respect to $\mathbb{C}_i$.

Since this is true of every $P_1 \in \mathbb{D}_1$, and the star number and VC dimension of $\mathbb{C}_1$ are both equal d, the upper bound in Theorem 4 implies $\forall \varepsilon \in (0, (1-2\beta)/8)$, $\delta \in (0, 1/8]$,

$$\Lambda_{1,0}\left(\frac{\varepsilon}{3}, \frac{\delta}{3}\right) \lesssim \frac{1}{(1-2\beta)^2} d \cdot \mathrm{polylog}\left(\frac{d}{\varepsilon\delta}\right).$$

Similarly, since every $P_2 \in \mathbb{D}_2$ satisfies the bounded noise condition (with parameter β) with respect to $\mathbb{C}_2$, and the star number of $\mathbb{C}_2$ is $\mathfrak{s} - d \leq \mathfrak{s}$ while the VC dimension of $\mathbb{C}_2$ is 1, the upper bound in Theorem 4 implies $\forall \varepsilon \in (0, (1-2\beta)/8)$, $\delta \in (0, 1/8]$,

$$\Lambda_{2,0}\left(\frac{\varepsilon}{3}, \frac{\delta}{3}\right) \lesssim \frac{1}{(1-2\beta)^2} \min\left\{\mathfrak{s}, \frac{1-2\beta}{\varepsilon}\right\} \mathrm{polylog}\left(\frac{1}{\varepsilon\delta}\right).$$

Altogether, we have that

$$\Lambda_{\mathrm{BN}(\beta)}(\varepsilon, \delta) \lesssim \frac{1}{(1-2\beta)^2} \max\left\{\min\left\{\mathfrak{s}, \frac{1-2\beta}{\varepsilon}\right\}, d\right\} \mathrm{polylog}\left(\frac{d}{\varepsilon\delta}\right).$$

17. The term $\mathrm{Log}\left(\min\left\{\frac{1}{\varepsilon}, |\mathbb{C}|\right\}\right)$ in the lower bound is dominated by the other terms in this example, so that this upper bound is still consistent with the existence of this term in the lower bound.

For β bounded away from 0, this is within logarithmic factors of the lower bound in Theorem 4, so that we may conclude that the lower bound is sometimes tight to within logarithmic factors in this case. Furthermore, when β is near 0, it is within logarithmic factors of the lower bound in Theorem 3, which is also a lower bound on $\Lambda_{\mathrm{BN}(\beta)}(\varepsilon,\delta)$ since $\mathrm{RE} \subseteq \mathrm{BN}(\beta)$; thus, this inherited lower bound on $\Lambda_{\mathrm{BN}(\beta)}(\varepsilon,\delta)$ is also sometimes tight to within logarithmic factors when β is near 0.

D.2.3 Tsybakov Noise

The case of Tsybakov noise is slightly more involved than the above. In this case, fix any $a \in [1,\infty)$, $\alpha \in (0,1)$. Since the upper bound in Theorem 5 already matches the lower bound up to logarithmic factors when $\alpha \in (0,1/2]$, it suffices to focus on the case $\alpha \in (1/2,1)$. In this case, for $\gamma \in (0,1]$, let $\mathbb{D}_i(\gamma)$ denote the set of all $P_i \in \mathrm{TN}(a/\gamma^{1-\alpha},\alpha)$ with $P_i(\mathcal{X}_i \times \mathcal{Y}) = 1$, for each $i \in \{1,2\}$. Also let $\mathbb{D}_i(0)$ denote the set of all probability measures P_i with $P_i(\mathcal{X}_i \times \mathcal{Y}) = 1$, for each $i \in \{1,2\}$. Again, for any $P \in \mathrm{TN}(a,\alpha)$, $P(\cdot) = P(\mathcal{X}_1 \times \mathcal{Y})P(\cdot|\mathcal{X}_1 \times \mathcal{Y}) + P(\mathcal{X}_2 \times \mathcal{Y})P(\cdot|\mathcal{X}_2 \times \mathcal{Y})$, and for any $i \in \{1,2\}$ with $P(\mathcal{X}_i \times \mathcal{Y}) > 0$, $P(\cdot \times \mathcal{Y}|\mathcal{X}_i \times \mathcal{Y})$ is supported only in $\mathcal{X}_i$, and $\eta(\cdot; P(\cdot|\mathcal{X}_i \times \mathcal{Y})) = \eta(\cdot;P)$ on $\mathcal{X}_i$, so that for any $t > 0$,

$$
\begin{aligned}
&P\left(\{x : |\eta(x; P(\cdot|\mathcal{X}_i \times \mathcal{Y})) - 1/2| \le t\} \times \mathcal{Y} \middle| \mathcal{X}_i \times \mathcal{Y}\right) \\
&= \frac{1}{P(\mathcal{X}_i \times \mathcal{Y})} P\left(\{x \in \mathcal{X}_i : |\eta(x;P) - 1/2| \le t\} \times \mathcal{Y}\right) \\
&\le \frac{1}{P(\mathcal{X}_i \times \mathcal{Y})} a' t^{\alpha/(1-\alpha)} = (1-\alpha)(2\alpha)^{\alpha/(1-\alpha)} \left(\frac{a}{P(\mathcal{X}_i \times \mathcal{Y})^{1-\alpha}}\right)^{1/(1-\alpha)} t^{\alpha/(1-\alpha)}.
\end{aligned}
$$

Also, since $f_P^\star \in \mathbb{C}$, and $\eta(\cdot; P(\cdot|\mathcal{X}_i \times \mathcal{Y})) = \eta(\cdot;P)$ on $\mathcal{X}_i$, we can take $f^\star_{P(\cdot|\mathcal{X}_i \times \mathcal{Y})}(x) = f_P^\star(x)$ for every $x \in \mathcal{X}_i$, so that there exists a version of $f^\star_{P(\cdot|\mathcal{X}_i \times \mathcal{Y})}$ contained in $\mathbb{C}$. Together, these imply that $P(\cdot|\mathcal{X}_i \times \mathcal{Y}) \in \mathbb{D}_i(P(\mathcal{X}_i \times \mathcal{Y}))$. We therefore have that $\forall P \in \mathrm{TN}(a,\alpha)$, $P = \gamma P_1 + (1-\gamma)P_2$ for some $\gamma \in [0,1]$, $P_1 \in \mathbb{D}_1(\gamma)$, and $P_2 \in \mathbb{D}_2(1-\gamma)$: that is, $\mathrm{TN}(a,\alpha) \subseteq \mathbb{D}$, for $\mathbb{D}$ as in Lemma 46 (with respect to these definitions of $\mathbb{D}_i(\cdot)$). Therefore, Lemma 46 implies that $\forall \varepsilon,\delta \in (0,1)$,

$$
\begin{aligned}
&\Lambda_{\mathrm{TN}(a,\alpha)}(\varepsilon,\delta) \le \Lambda_{\mathbb{D}}(\varepsilon,\delta) \\
&\lesssim \sup_{\gamma \in [0,1]} \max\left\{\Lambda_{1,(\gamma-\varepsilon/8)\vee 0}\left(\frac{\varepsilon}{2(\gamma+\varepsilon/8)}, \frac{\delta}{3}\right), \Lambda_{2,(1-\gamma-\varepsilon/8)\vee 0}\left(\frac{\varepsilon}{2(1-\gamma+\varepsilon/8)}, \frac{\delta}{3}\right)\right\}. \quad (68)
\end{aligned}
$$

First note that, for the case $\gamma \le \varepsilon/4$, we trivially have

$$
\Lambda_{1,(\gamma-\varepsilon/8)\vee 0}\left(\frac{\varepsilon}{2(\gamma+\varepsilon/8)}, \frac{\delta}{3}\right) \le \Lambda_{1,0}\left(\frac{\varepsilon}{2(\gamma+\varepsilon/4)}, \frac{\delta}{3}\right) \le \Lambda_{1,0}\left(1, \frac{\delta}{3}\right) = 0,
$$

and similarly for the case $\gamma \ge 1 - \varepsilon/4$, we have $\Lambda_{2,(1-\gamma-\varepsilon/8)\vee 0}\left(\frac{\varepsilon}{2(1-\gamma+\varepsilon/8)}, \frac{\delta}{3}\right) = 0$.

For the remaining cases, for any $\gamma \in (0,1]$, since every $P_i \in \mathbb{D}_i(\gamma)$ has $f^\star_{P_i} \in \mathbb{C}$, and every $h \in \mathbb{C} \setminus \mathbb{C}_i$ has $h(x) = -1$ for every $x \in \mathcal{X}_i$, and the all-negative function $x \mapsto -1$ is contained in $\mathbb{C}_i$, and $P_i(\mathcal{X}_i \times \mathcal{Y}) = 1$, without loss we can take $f^\star_{P_i} \in \mathbb{C}_i$. Together with

the definition of $\mathbb{D}_i(\gamma)$, we have that $\mathbb{D}_i(\gamma)$ is contained in the set of probability measures P_i satisfying the Tsybakov noise condition with respect to the hypothesis class $\mathbb{C}_i$, with parameters $\frac{a}{\gamma^{1-\alpha}}$ and α. Therefore, since the star number and VC dimension of $\mathbb{C}_1$ are both d, Theorem 5 implies that for any $\gamma \in (\varepsilon/4, 1]$,[18]

$$\Lambda_{1,\gamma-\varepsilon/8}\left(\frac{\varepsilon}{2(\gamma+\varepsilon/8)}, \frac{\delta}{3}\right) \leq \Lambda_{1,\gamma/2}\left(\frac{\varepsilon}{3\gamma}, \frac{\delta}{3}\right)$$
$$\lesssim \left(\frac{a}{\gamma^{1-\alpha}}\right)^2 \left(\frac{\gamma}{\varepsilon}\right)^{2-2\alpha} d \cdot \operatorname{polylog}\left(\frac{d}{\varepsilon\delta}\right) = a^2 \left(\frac{1}{\varepsilon}\right)^{2-2\alpha} d \cdot \operatorname{polylog}\left(\frac{d}{\varepsilon\delta}\right).$$

Similarly, since the star number of $\mathbb{C}_2$ is $\mathfrak{s} - d$ and the VC dimension of $\mathbb{C}_2$ is 1, Theorem 5 implies that for any $\gamma \in [0, 1-\varepsilon/4)$,

$$\Lambda_{2,1-\gamma-\varepsilon/8}\left(\frac{\varepsilon}{2(1-\gamma+\varepsilon/8)}, \frac{\delta}{3}\right) \leq \Lambda_{2,(1-\gamma)/2}\left(\frac{\varepsilon}{3(1-\gamma)}, \frac{\delta}{3}\right)$$
$$\lesssim \left(\frac{a}{(1-\gamma)^{1-\alpha}}\right)^2 \left(\frac{1-\gamma}{\varepsilon}\right)^{2-2\alpha} \min\left\{\mathfrak{s} - d, \frac{(1-\gamma)^{1/\alpha}(1-\gamma)}{a^{1/\alpha}\varepsilon}\right\}^{2\alpha-1} \operatorname{polylog}\left(\frac{1}{\varepsilon\delta}\right)$$
$$\leq a^2 \left(\frac{1}{\varepsilon}\right)^{2-2\alpha} \min\left\{\mathfrak{s}, \frac{1}{a^{1/\alpha}\varepsilon}\right\}^{2\alpha-1} \operatorname{polylog}\left(\frac{1}{\varepsilon\delta}\right).$$

Plugging this into (68), we have that

$$\Lambda_{\mathrm{TN}(a,\alpha)}(\varepsilon, \delta) \lesssim a^2 \left(\frac{1}{\varepsilon}\right)^{2-2\alpha} \max\left\{\min\left\{\mathfrak{s}, \frac{1}{a^{1/\alpha}\varepsilon}\right\}^{2\alpha-1}, d\right\} \operatorname{polylog}\left(\frac{d}{\varepsilon\delta}\right).$$

As claimed, this is within logarithmic factors of the lower bound in Theorem 5 (for $1/2 < \alpha < 1$, $a \geq 4$, $\varepsilon \in (0, 1/(24a^{1/\alpha}))$, and $\delta \in (0, 1/24]$), so that, combined with the tightness (always) for the case $0 < \alpha \leq 1/2$, we may conclude that the lower bounds in Theorem 5 are sometimes tight to within logarithmic factors.

D.2.4 Benign Noise

The case of benign noise proceeds analogously to the above. Since $\mathrm{BE}(0) = \mathrm{RE}$, tightness of the lower bound for the case $\nu = 0$ (up to constant factors) has already been addressed above (supposing we include the lower bound from Theorem 3 as a lower bound on $\Lambda_{\mathrm{BE}(\nu)}(\varepsilon, \delta)$ to strengthen the lower bound in Theorem 7). For the remainder, we suppose $\nu \in (0, 1/2)$. For $\gamma \in [0, 1]$, let $\mathbb{D}_i(\gamma)$ denote the set of all $P_i \in \mathrm{BE}(\nu/(\gamma \vee 2\nu))$ with $P_i(\mathcal{X}_i \times \mathcal{Y}) = 1$, for each $i \in \{1, 2\}$. Again, for any $P \in \mathrm{BE}(\nu)$, $P(\cdot) = P(\mathcal{X}_1 \times \mathcal{Y})P(\cdot|\mathcal{X}_1 \times \mathcal{Y}) + P(\mathcal{X}_2 \times \mathcal{Y})P(\cdot|\mathcal{X}_2 \times \mathcal{Y})$, and for any $i \in \{1, 2\}$ with $P(\mathcal{X}_i \times \mathcal{Y}) > 0$, $P(\cdot \times \mathcal{Y}|\mathcal{X}_i \times \mathcal{Y})$ is supported only in $\mathcal{X}_i$, and $\eta(\cdot; P(\cdot|\mathcal{X}_i \times \mathcal{Y})) = \eta(\cdot; P)$ on $\mathcal{X}_i$, so that we can take $f^\star_{P(\cdot|\mathcal{X}_i \times \mathcal{Y})}(x) = f^\star_P(x)$ for every $x \in \mathcal{X}_i$;

18. Recall that, as mentioned in Section 5, the upper bounds on the label complexities stated in Section 5 hold without the stated restrictions on the values $\varepsilon, \delta \in (0, 1)$ and a.

thus, there is a version of $f^\star_{P(\cdot|\mathcal{X}_i\times\mathcal{Y})}$ contained in $\mathbb{C}$. Furthermore,

$$\begin{aligned}\mathrm{er}_{P(\cdot|\mathcal{X}_i\times\mathcal{Y})}(f^\star_{P(\cdot|\mathcal{X}_i\times\mathcal{Y})}) &= \frac{1}{P(\mathcal{X}_i\times\mathcal{Y})}P\left((x,y): f^\star_P(x)\neq y \text{ and } x\in\mathcal{X}_i\right)\\ &\le \frac{1}{P(\mathcal{X}_i\times\mathcal{Y})}P\left((x,y): f^\star_P(x)\neq y\right) \le \frac{\nu}{P(\mathcal{X}_i\times\mathcal{Y})}.\end{aligned}$$

Also, since every $x\in\mathcal{X}_i$ has $f^\star_{P(\cdot|\mathcal{X}_i\times\mathcal{Y})}(x) = f^\star_P(x) = \mathrm{sign}(2\eta(x;P)-1) = \mathrm{sign}(2\eta(x;P(\cdot|\mathcal{X}_i\times\mathcal{Y}))-1)$, we have $P((x,y): f^\star_{P(\cdot|\mathcal{X}_i\times\mathcal{Y})}(x)=y|x\in\mathcal{X}_i)\ge 1/2$, so that $\mathrm{er}_{P(\cdot|\mathcal{X}_i\times\mathcal{Y})}(f^\star_{P(\cdot|\mathcal{X}_i\times\mathcal{Y})})\le 1/2$. Together, these imply that $P(\cdot|\mathcal{X}_i\times\mathcal{Y})\in\mathbb{D}_i(P(\mathcal{X}_i\times\mathcal{Y}))$. We therefore have that $\forall P\in\mathrm{BE}(\nu)$, $P=\gamma P_1+(1-\gamma)P_2$ for some $\gamma\in[0,1]$, $P_1\in\mathbb{D}_1(\gamma)$, and $P_2\in\mathbb{D}_2(1-\gamma)$: that is, $\mathrm{BE}(\nu)\subseteq\mathbb{D}$, for $\mathbb{D}$ as in Lemma 46 (with respect to these definitions of $\mathbb{D}_i(\cdot)$). Therefore, Lemma 46 implies that $\forall\varepsilon,\delta\in(0,1)$,

$$\begin{aligned}&\Lambda_{\mathrm{BE}(\nu)}(\varepsilon,\delta)\le\Lambda_{\mathbb{D}}(\varepsilon,\delta)\\ &\lesssim \sup_{\gamma\in[0,1]}\max\left\{\Lambda_{1,(\gamma-\varepsilon/8)\vee 0}\left(\frac{\varepsilon}{2(\gamma+\varepsilon/8)},\frac{\delta}{3}\right),\Lambda_{2,(1-\gamma-\varepsilon/8)\vee 0}\left(\frac{\varepsilon}{2(1-\gamma+\varepsilon/8)},\frac{\delta}{3}\right)\right\}. \quad (69)\end{aligned}$$

First note that, as above, for the case $\gamma\le\varepsilon/4$, we trivially have

$$\Lambda_{1,(\gamma-\varepsilon/8)\vee 0}\left(\frac{\varepsilon}{2(\gamma+\varepsilon/8)},\frac{\delta}{3}\right)\le\Lambda_{1,0}\left(\frac{\varepsilon}{2(\gamma+\varepsilon/4)},\frac{\delta}{3}\right)\le\Lambda_{1,0}\left(1,\frac{\delta}{3}\right)=0,$$

and similarly for the case $\gamma\ge 1-\varepsilon/4$, we have $\Lambda_{2,(1-\gamma-\varepsilon/8)\vee 0}\left(\frac{\varepsilon}{2(1-\gamma+\varepsilon/8)},\frac{\delta}{3}\right)=0$.

For the remaining cases, for any $\gamma\in(0,1]$, since every $P_i\in\mathbb{D}_i(\gamma)$ has $f^\star_{P_i}\in\mathbb{C}$, and every $h\in\mathbb{C}\setminus\mathbb{C}_i$ has $h(x)=-1$ for every $x\in\mathcal{X}_i$, and the all-negative function $x\mapsto -1$ is contained in $\mathbb{C}_i$, and $P_i(\mathcal{X}_i\times\mathcal{Y})=1$, without loss we can take $f^\star_{P_i}\in\mathbb{C}_i$. Together with the definition of $\mathbb{D}_i(\gamma)$, we have that $\mathbb{D}_i(\gamma)$ is contained in the set of probability measures P_i satisfying the benign noise condition with respect to the hypothesis class $\mathbb{C}_i$, with parameter $\frac{\nu}{\gamma}\wedge\frac{1}{2}$. Therefore, since the star number and VC dimension of $\mathbb{C}_1$ are both d, Theorem 7 implies that for any $\gamma\in(\varepsilon/4,1]$,[19]

$$\begin{aligned}\Lambda_{1,\gamma-\varepsilon/8}\left(\frac{\varepsilon}{2(\gamma+\varepsilon/8)},\frac{\delta}{3}\right)\le\Lambda_{1,\gamma/2}\left(\frac{\varepsilon}{3\gamma},\frac{\delta}{3}\right)&\lesssim\left(\frac{(\nu/\gamma)^2}{(\varepsilon/\gamma)^2}d+d\right)\mathrm{polylog}\left(\frac{d}{\varepsilon\delta}\right)\\ &\lesssim\left(\frac{\nu^2}{\varepsilon^2}\vee 1\right)d\cdot\mathrm{polylog}\left(\frac{d}{\varepsilon\delta}\right).\end{aligned}$$

Similarly, since the star number of $\mathbb{C}_2$ is $\mathfrak{s}-d$ and the VC dimension of $\mathbb{C}_2$ is 1, Theorem 7 implies that for any $\gamma\in[0,1-\varepsilon/4)$,

$$\begin{aligned}&\Lambda_{2,1-\gamma-\varepsilon/8}\left(\frac{\varepsilon}{2(1-\gamma+\varepsilon/8)},\frac{\delta}{3}\right)\le\Lambda_{2,(1-\gamma)/2}\left(\frac{\varepsilon}{3(1-\gamma)},\frac{\delta}{3}\right)\\ &\lesssim\left(\frac{(\nu/(1-\gamma))^2}{(\varepsilon/(1-\gamma))^2}+\min\left\{\mathfrak{s}-d,\frac{1}{\varepsilon}\right\}\right)\mathrm{polylog}\left(\frac{1}{\varepsilon\delta}\right)\lesssim\left(\frac{\nu^2}{\varepsilon^2}\vee\min\left\{\mathfrak{s},\frac{1}{\varepsilon}\right\}\right)\mathrm{polylog}\left(\frac{1}{\varepsilon\delta}\right).\end{aligned}$$

19. Again, as mentioned in Section 5, the restrictions on ε,δ stated in Theorem 7 are only required for the lower bounds.

Plugging these into (69), we have that for $\varepsilon \in (0, \nu)$,

$$\Lambda_{\mathrm{BE}(\nu)}(\varepsilon, \delta) \lesssim \left(\frac{\nu^2}{\varepsilon^2} d + \min\left\{ \mathfrak{s}, \frac{1}{\varepsilon} \right\} \right) \mathrm{polylog}\left(\frac{d}{\varepsilon\delta} \right).$$

Again, this is within logarithmic factors of the lower bound in Theorem 7 (for $\varepsilon \in (0, (1-2\nu)/24)$ and $\delta \in (0, 1/24]$), so that we may conclude that this lower bound is sometimes tight to within logarithmic factors when ν is not near 0 (specifically, when $\varepsilon < \nu$). For $\nu \leq \varepsilon$, the above implies

$$\Lambda_{\mathrm{BE}(\nu)}(\varepsilon, \delta) \lesssim \max\left\{ d, \min\left\{ \mathfrak{s}, \frac{1}{\varepsilon} \right\} \right\} \mathrm{polylog}\left(\frac{d}{\varepsilon\delta} \right),$$

which is within logarithmic factors of the lower bound in Theorem 3 (for $\varepsilon \in (0, 1/9)$ and $\delta \in (0, 1/3)$). Since $\mathrm{RE} \subseteq \mathrm{BE}(\nu)$, this is also a lower bound on $\Lambda_{\mathrm{BE}(\nu)}(\varepsilon, \delta)$. Thus, in this case, we may conclude that this inherited lower bound on $\Lambda_{\mathrm{BE}(\nu)}(\varepsilon, \delta)$ is sometimes tight to within logarithmic factors, for ν near 0 (specifically, when $\varepsilon \geq \nu$).

References

T. M. Adams and A. B. Nobel. Uniform convergence of Vapnik-Chervonenkis classes under ergodic sampling. *Annals of Probability*, 38(4):1345–1367, 2010.

T. M. Adams and A. B. Nobel. Uniform approximation and bracketing properties of VC classes. *Bernoulli*, 18:1310–1319, 2012.

N. Ailon, R. Begleiter, and E. Ezra. Active learning using smooth relative regret approximations with applications. In *Proceedings of the* 25th *Conference on Learning Theory*, 2012.

M. Anthony and P. Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, 1999.

P. Awasthi, M.-F. Balcan, and P. M. Long. The power of localization for efficiently learning linear separators with noise. In *Proceedings of the* 46th *ACM Symposium on the Theory of Computing*, 2014.

M.-F. Balcan and S. Hanneke. Robust interactive learning. In *Proceedings of the* 25th *Conference on Learning Theory*, 2012.

M.-F. Balcan and P. M. Long. Active and passive learning of linear separators under log-concave distributions. In *Proceedings of the* 26th *Conference on Learning Theory*, 2013.

M.-F. Balcan, A. Beygelzimer, and J. Langford. Agnostic active learning. In *Proceedings of the* 23rd *International Conference on Machine Learning*, 2006.

M.-F. Balcan, A. Broder, and T. Zhang. Margin based active learning. In *Proceedings of the* 20th *Conference on Learning Theory*, 2007.

M.-F. Balcan, A. Beygelzimer, and J. Langford. Agnostic active learning. *Journal of Computer and System Sciences*, 75(1):78–89, 2009.

M.-F. Balcan, S. Hanneke, and J. Wortman Vaughan. The true sample complexity of active learning. *Machine Learning*, 80(2–3):111–139, 2010.

P. Bartlett, S. Mendelson, and P. Philips. Local complexities for empirical risk minimization. In *Proceedings of the* 17th *Conference on Learning Theory*, 2004.

P. Bartlett, M. I. Jordan, and J. McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101:138–156, 2006.

A. Beygelzimer, S. Dasgupta, and J. Langford. Importance weighted active learning. In *Proceedings of the* 26th *International Conference on Machine Learning*, 2009.

A. Beygelzimer, D. Hsu, J. Langford, and T. Zhang. Agnostic active learning without constraints. In *Advances in Neural Information Processing Systems* 23, 2010.

A. Blumer, A. Ehrenfeucht, D. Haussler, and M. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the Association for Computing Machinery*, 36(4): 929–965, 1989.

S. Boucheron, O. Bousquet, and G. Lugosi. Theory of classification: A survey of some recent advances. *ESAIM: Probability and Statistics*, 9:323–375, November 2005.

O. Bousquet, S. Boucheron, and G. Lugosi. Introduction to statistical learning theory. *Lecture Notes in Artificial Intelligence*, 3176:169–207, 2004.

N. H. Bshouty, Y. Li, and P. M. Long. Using the doubling dimension to analyze the generalization of learning algorithms. *Journal of Computer and System Sciences*, 75(6): 323–335, 2009.

R. M. Castro and R. D. Nowak. Upper and lower error bounds for active learning. In *The* 44th *Annual Allerton Conference on Communication, Control and Computing*, 2006.

R. M. Castro and R. D. Nowak. Minimax bounds for active learning. *IEEE Transactions on Information Theory*, 54(5):2339–2353, July 2008.

D. Cohn, L. Atlas, and R. Ladner. Improving generalization with active learning. *Machine Learning*, 15(2):201–221, 1994.

S. Dasgupta. Analysis of a greedy active learning strategy. In *Advances in Neural Information Processing Systems* 17, 2004.

S. Dasgupta. Coarse sample complexity bounds for active learning. In *Advances in Neural Information Processing Systems* 18, 2005.

S. Dasgupta, A. T. Kalai, and C. Monteleoni. Analysis of perceptron-based active learning. In *Proceedings of the* 18th *Conference on Learning Theory*, 2005.

S. Dasgupta, D. Hsu, and C. Monteleoni. A general agnostic active learning algorithm. In *Advances in Neural Information Processing Systems* 20, 2007.

A. Ehrenfeucht, D. Haussler, M. Kearns, and L. Valiant. A general lower bound on the number of examples needed for learning. *Information and Computation*, 82:247–261, 1989.

R. El-Yaniv and Y. Wiener. On the foundations of noise-free selective classification. *Journal of Machine Learning Research*, 11:1605–1641, 2010.

R. El-Yaniv and Y. Wiener. Active learning via perfect selective classification. *Journal of Machine Learning Research*, 13:255–279, 2012.

G. Fan. A graph-theoretic view of teaching. Master's thesis, Department of Computer Science, University of Regina, 2012.

S. Floyd and M. Warmuth. Sample compression, learnability, and the Vapnik-Chervonenkis dimension. *Machine Learning*, 21:269–304, 1995.

Y. Freund, H. S. Seung, E. Shamir, and N. Tishby. Selective sampling using the query by committee algorithm. *Machine Learning*, 28:133–168, 1997.

E. Friedman. Active learning for smooth problems. In *Proceedings of the* 22nd *Conference on Learning Theory*, 2009.

E. Giné and V. Koltchinskii. Concentration inequalities and asymptotic results for ratio type empirical processes. *The Annals of Probability*, 34(3):1143–1216, 2006.

S. A. Goldman and M. J. Kearns. On the complexity of teaching. *Journal of Computer and System Sciences*, 50:20–31, 1995.

A. Gupta, R. Krauthgamer, and J. R. Lee. Bounded geometries, fractals, and low-distortion embeddings. In *Proceedings of the* 44th *Annual IEEE Symposium on Foundations of Computer Science*, 2003.

S. Hanneke. The cost complexity of interactive learning. *Unpublished manuscript*, 2006.

S. Hanneke. Teaching dimension and the complexity of active learning. In *Proceedings of the* 20th *Conference on Learning Theory*, 2007a.

S. Hanneke. A bound on the label complexity of agnostic active learning. In *Proceedings of the* 24th *International Conference on Machine Learning*, 2007b.

S. Hanneke. Adaptive rates of convergence in active learning. In *Proceedings of the* 22nd *Conference on Learning Theory*, 2009a.

S. Hanneke. *Theoretical Foundations of Active Learning*. PhD thesis, Machine Learning Department, School of Computer Science, Carnegie Mellon University, 2009b.

S. Hanneke. Rates of convergence in active learning. *The Annals of Statistics*, 39(1):333–361, 2011.

S. Hanneke. Activized learning: Transforming passive to active with improved label complexity. *Journal of Machine Learning Research*, 13(5):1469–1587, 2012.

S. Hanneke. Theory of Active Learning. Version 1.1. `http://www.stevehanneke.com`, 2014.

S. Hanneke and L. Yang. Surrogate losses in passive and active learning. *arXiv:1207.3772*, 2012.

D. Haussler and E. Welzl. ε-nets and simplex range queries. *Discrete Computational Geometry*, 2:127–151, 1987.

D. Haussler, N. Littlestone, and M. Warmuth. Predicting $\{0,1\}$-functions on randomly drawn points. *Information and Computation*, 115:248–292, 1994.

T. Hegedüs. Generalized teaching dimensions and the query complexity of learning. In *Proceedings of the* 8^{th} *Conference on Computational Learning Theory*, 1995.

L. Hellerstein, K. Pillaipakkamnatt, V. Raghavan, and D. Wilkins. How many queries are needed to learn? *Journal of the Association for Computing Machinery*, 43(5):840–862, 1996.

D. Hsu. *Algorithms for Active Learning*. PhD thesis, Department of Computer Science and Engineering, School of Engineering, University of California, San Diego, 2010.

M. Kääriäinen. Active learning in the non-realizable case. In *Proceedings of the* 17^{th} *International Conference on Algorithmic Learning Theory*, 2006.

A. T. Kalai, A. R. Klivans, Y. Mansour, and R. A. Servedio. Agnostically learning halfspaces. In *Proceedings of the* 46^{th} *Annual IEEE Symposium on Foundations of Computer Science*, 2005.

O. Kallenberg. *Foundations of Modern Probability,* 2^{nd} *Edition*. Springer Verlag, New York, 2002.

M. J. Kearns, R. E. Schapire, and L. M. Sellie. Toward efficient agnostic learning. *Machine Learning*, 17:115–141, 1994.

A. N. Kolmogorov and V. M. Tikhomirov. ε-entropy and ε-capacity of sets in functional spaces. *Uspekhi Matematicheskikh Nauk*, 14(2):3–86, 1959.

A. N. Kolmogorov and V. M. Tikhomirov. ε-entropy and ε-capacity of sets in functional spaces. *American Mathematical Society Translations, Series* 2, 17:277–364, 1961.

V. Koltchinskii. Local Rademacher complexities and oracle inequalities in risk minimization. *The Annals of Statistics*, 34(6):2593–2656, 2006.

V. Koltchinskii. Rademacher complexities and bounding the excess risk in active learning. *Journal of Machine Learning Research*, 11:2457–2485, 2010.

S. R. Kulkarni. On metric entropy, Vapnik-Chervonenkis dimension, and learnability for a class of distributions. Technical Report CICS-P-160, Center for Intelligent Control Systems, 1989.

S. R. Kulkarni, S. K. Mitter, and J. N. Tsitsiklis. Active learning using arbitrary binary valued queries. *Machine Learning*, 11:23–35, 1993.

L. LeCam. Convergence of estimates under dimensionality restrictions. *The Annals of Statistics*, 1(1):38–53, 1973.

Y. Li and P. M. Long. Learnability and the doubling dimension. In *Advances in Neural Information Processing* 20, 2007.

N. Littlestone and M. Warmuth. Relating data compression and learnability. *Unpublished manuscript*, 1986.

E. Mammen and A.B. Tsybakov. Smooth discrimination analysis. *The Annals of Statistics*, 27:1808–1829, 1999.

P. Massart and E. Nédélec. Risk bounds for statistical learning. *The Annals of Statistics*, 34(5):2326–2366, 2006.

C. McDiarmid. Concentration. In *Probabilistic Methods for Algorithmic Discrete Mathematics*, pages 195–248. Springer-Verlag, 1998.

S. Minsker. Plug-in approach to active learning. *Journal of Machine Learning Research*, 13 (1):67–90, 2012.

T. Mitchell. Version spaces: A candidate elimination approach to rule learning. In *Proceedings of the* 5^{th} *International Joint Conference on Artificial Intelligence*, 1977.

M. Raginsky and A. Rakhlin. Lower bounds for passive and active learning. In *Advances in Neural Information Processing Systems* 24, 2011.

N. Sauer. On the density of families of sets. *Journal of Combinatorial Theory (A)*, 13: 145–147, 1972.

B. Settles. *Active Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan & Claypool Publishers, 2012.

A. B. Tsybakov. Optimal aggregation of classifiers in statistical learning. *The Annals of Statistics*, 32(1):135–166, 2004.

L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, November 1984.

A. W. van der Vaart and J. A. Wellner. *Weak Convergence and Empirical Processes*. Springer, 1996.

A. W. van der Vaart and J. A. Wellner. A local maximal inequality under uniform entropy. *Electronic Journal of Statistics*, 5:192–203, 2011.

R. van Handel. The universal Glivenko-Cantelli property. *Probability and Related Fields*, 155:911–934, 2013.

V. Vapnik. *Estimation of Dependencies Based on Empirical Data*. Springer-Verlag, New York, 1982.

V. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, Inc., New York, 1998.

V. Vapnik and A. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16:264–280, 1971.

V. Vapnik and A. Chervonenkis. *Theory of Pattern Recognition*. Nauka, Moscow, 1974.

M. Vidyasagar. *Learning and Generalization with Applications to Neural Networks,* 2nd *Edition*. Springer-Verlag, 2003.

J. von Neumann. Zur theorie der gesellschaftsspiele. *Mathematische Annalen*, 100(1):295–320, 1928.

J. von Neumann and O. Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 1944.

A. Wald. Sequential tests of statistical hypotheses. *The Annals of Mathematical Statistics*, 16(2):117–186, 1945.

A. Wald. *Sequential Analysis*. John Wiley & Sons, Inc., New York, 1947.

L. Wang. Smoothness, disagreement coefficient, and the label complexity of agnostic active learning. *Journal of Machine Learning Research*, 12:2269–2292, 2011.

Y. Wiener, S. Hanneke, and R. El-Yaniv. A compression technique for analyzing disagreement-based active learning. *Journal of Machine Learning Research*, 16(4):713–745, 2015.

Y. Yang and A. Barron. Information-theoretic determination of minimax rates of convergence. *The Annals of Statistics*, 27(5):1564–1599, 1999.

Journal of Machine Learning Research 16 (2015) 3603-3635 Submitted 7/14; Revised 1/15; Published 12/15

Convergence Rates for Persistence Diagram Estimation in Topological Data Analysis

Frédéric Chazal FREDERIC.CHAZAL@INRIA.FR
Marc Glisse MARC.GLISSE@INRIA.FR
Inria Saclay - Île de France
1 rue Honoré d'Estienne d'Orves
Bâtiment Alan Turing
Campus de l'École Polytechnique
91120 Palaiseau, France

Catherine Labruère CLABRUER@U-BOURGOGNE.FR
Université de Bourgogne
Institut de Mathématiques de Bourgogne, UMR CNRS 5584
9 Avenue Alain Savary, B.P. 47870
21078 Dijon, France

Bertrand Michel BERTRAND.MICHEL@UPMC.FR
LSTA, Université Pierre et Marie Curie Paris 6
15-25, bureau 220
4, place Jussieu
75005 Paris, France

Editor: Matthias Hein

Abstract

Computational topology has recently seen an important development toward data analysis, giving birth to the field of topological data analysis. Topological persistence, or persistent homology, appears as a fundamental tool in this field. In this paper, we study topological persistence in general metric spaces, with a statistical approach. We show that the use of persistent homology can be naturally considered in general statistical frameworks and that persistence diagrams can be used as statistics with interesting convergence properties. Some numerical experiments are performed in various contexts to illustrate our results.

Keywords: persistent homology, convergence rates, topological data analysis

1. Introduction

During the last decades, the wide availability of measurement devices and simulation tools has led to an explosion in the amount of available data in almost all domains of science, industry, economy and even everyday life. Often these data come as point clouds sampled in possibly high (or infinite) dimensional spaces. They are usually not uniformly distributed in the embedding space but carry some geometric structure (manifold or more general stratified space) which reflects important properties of the "systems" from which they have been generated. Moreover, in many cases data are not embedded in Euclidean spaces and come as (finite) sets of points with pairwise distance information. This often happens,

e.g. with social network or sensor network data where each sensor may not know its own position, but may evaluate its distance to the other sensors using the strength of the signal received from them. In such cases, data are given as matrices of pairwise distances between the observations, i.e. as (discrete) metric spaces. Again, although they come as abstract spaces, these data often carry specific topological and geometric structures.

1.1 Topological Data Analysis

A large amount of research has been done on dimensionality reduction, manifold learning and geometric inference for data embedded in Euclidean spaces and assumed to be concentrated around submanifolds; see for instance Wang (2012). However, the assumption that data lies on a manifold may fail in many applications. In addition, the strategy of representing data by points in Euclidean spaces may introduce large metric distortions as the data may lie in highly curved spaces. With the emergence of new geometric inference and algebraic topology tools, computational topology (Edelsbrunner and Harer, 2010) has recently seen an important development toward data analysis, giving birth to the field of Topological Data Analysis (TDA) (Carlsson, 2009) whose aim is to infer relevant, multiscale, qualitative and quantitative topological structures directly from the data. Topological persistence, more precisely *persistent homology* appears as a fundamental tool for TDA. Roughly, *homology* (with coefficient in a field such as, e.g., $\mathbb{Z}/2\mathbb{Z}$) associates to any topological space $\mathbb{M}$, a family of vector spaces (the so-called homology groups) $H_k(\mathbb{M})$, $k = 0, 1, \ldots$, each of them encoding topological features of $\mathbb{M}$. The k^{th} *Betti number* of $\mathbb{M}$, denoted β_k, is the dimension of $H_k(\mathbb{M})$ and measures the number of k-dimensional features of $\mathbb{M}$: for example, β_0 is the number of connected components of $\mathbb{M}$, β_1 the number of independent cycles or "tunnels", β_2 the number of "voids", etc. (see Hatcher, 2001). Persistent homology provides a framework (Edelsbrunner et al., 2002; Zomorodian and Carlsson, 2005; Chazal et al., 2012a) and efficient algorithms to encode the evolution of the homology of families of nested topological spaces indexed by a set of real numbers that may often be seen as scales, such as the sublevel sets of a function, the union of growing balls, etc. The obtained multiscale topological information is then represented in a simple way as a barcode or persistence diagram; see Figure 4 and Section 2.3.
In TDA, persistent homology has found applications in many fields, including neuroscience (Singh et al., 2008), bioinformatics (Kasson et al., 2007), shape classification (Chazal et al., 2009b), clustering (Chazal et al., 2013), sensor networks (De Silva and Ghrist, 2007) or signal processing (Bauer et al., 2014). It is usually computed for a *filtered simplicial complex* built on top of the available data, i.e. a nested family of simplicial complexes whose vertex set is the data set (see Section 2.3). The obtained persistence diagrams are then used as "topological signatures" to exhibit and compare the topological structure underlying the data; see Figure 1. The relevance of this approach relies on stability results ensuring that close data sets, with respect to the Hausdorff or Gromov-Hausdorff distance, have close persistence diagrams (Cohen-Steiner et al., 2007; Chazal et al., 2009a, 2012a,b). However these results are not statistical and thus only provide heuristic or exploratory uses in data analysis.

The goal of this paper is to show that, thanks to recent results by Chazal et al. (2012a,b) that allow to consider persistence diagrams associated to infinite spaces, the use of persis-

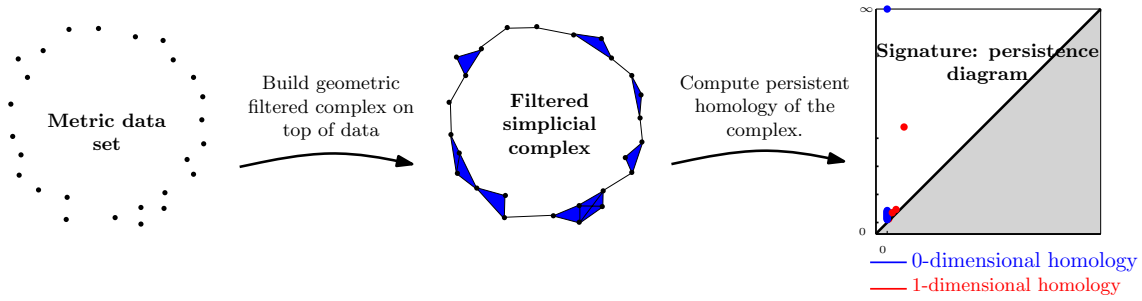


Figure 1: A classical pipeline for persistence in TDA.

tent homology in TDA can be naturally considered in general statistical frameworks and persistence diagrams can be used as statistics with interesting convergence properties.

1.2 Contribution

In this paper we assume that the available data is the realization of a probability distribution supported on an unknown compact metric space. We consider the persistent homology of different filtered simplicial complexes built on top of the data. We study, with a minimax approach, the rate of convergence of the associated persistence diagrams to some well-defined persistence diagram associated to the support of the probability distribution. More precisely, we assume that we observe a set of n points $\widehat{\mathbb{X}}_n = \{X_1 \dots, X_n\}$ in a metric space $(\mathbb{M}, \rho)$, drawn i.i.d. from some unknown measure μ whose support is a compact set denoted $\mathbb{X}_\mu \subseteq \mathbb{M}$. We also assume that μ satisfies the so-called (a,b)-*standard assumption* for some constants $a, b > 0$: for any $x \in \mathbb{X}_\mu$ and any $r > 0$, $\mu(B(x,r)) \geqslant \min(ar^b, 1)$. The following theorem illustrates the kind of results we obtain under such assumption.

Theorem (4 in Section 3): *Let $(\mathbb{M}, \rho)$, $a > 0$ and $b > 0$ as above. Then for any measure μ satisfying the (a,b)-standard assumption*

$$\mathbb{E}\left[\mathrm{d_b}(\mathsf{dgm}(\mathrm{Filt}(\mathbb{X}_\mu)), \mathsf{dgm}(\mathrm{Filt}(\widehat{\mathbb{X}}_n)))\right] \leqslant C\left(\frac{\ln n}{n}\right)^{1/b}$$

where the constant C only depends on a and b (not on $\mathbb{M}$). Assume moreover that there exists a non isolated point x in $\mathbb{M}$ and consider any sequence $(x_n) \in (\mathbb{M}\backslash\{x\})^{\mathbb{N}}$ such that $\rho(x, x_n) \leqslant (an)^{-1/b}$. Then for any estimator $\widehat{\mathsf{dgm}}_n$ of $\mathsf{dgm}(\mathrm{Filt}(\mathbb{X}_\mu))$:

$$\liminf_{n\to\infty} \rho(x, x_n)^{-1}\mathbb{E}\left[\mathrm{d_b}(\mathsf{dgm}(\mathrm{Filt}(\mathbb{X}_\mu)), \widehat{\mathsf{dgm}}_n)\right] \geqslant C'$$

where C' is an absolute constant.

Our approach relies on the general theory of persistence modules and our results follow from two recently proven properties of persistence diagrams (Chazal et al., 2012b, 2009a, 2012a).

First, as $\mathbb{X}_\mu$ can be any compact metric space (possibly infinite), the filtered complex $\mathrm{Filt}(\mathbb{X}_\mu)$ is usually not finite or even countable and the existence of its persistence diagram cannot be established from the "classical" persistence theory (Zomorodian and Carlsson, 2005; Edelsbrunner et al., 2002). In our setting, the existence of $\mathsf{dgm}(\mathrm{Filt}(\mathbb{X}_\mu))$ follows from

the general persistence framework introduced by Chazal et al. (2009a, 2012a). Notice that although this framework is rather abstract and theoretical, it does not have any practical drawback as only persistence diagrams of complexes built on top of finite data are computed. Second, a fundamental property of the persistence diagrams we are considering is their stability proven by Chazal et al. (2012b): the bottleneck distance between $\mathsf{dgm}(\mathrm{Filt}(\mathbb{X}_\mu))$ and $\mathsf{dgm}(\mathrm{Filt}(\widehat{\mathbb{X}}_n))$ is upper bounded by twice the Gromov-Hausdorff distance between $\mathbb{X}_\mu$ and $\widehat{\mathbb{X}}_n$. This result establishes a strong connection between our persistence estimation problem and support estimation problems. Upper bounds on the rate of convergence of persistence diagrams are then easily obtained using the same arguments as the ones usually used to obtain convergence results for support estimation with respect to the Hausdorff metric. We take advantage of this general remark to find rates of convergence of persistence diagrams in general metric spaces (Section 3) and also in the more classical case where the measure is supported in $\mathbb{R}^d$ (Section 4). Using Le Cam's lemma, we also compute the corresponding lower bounds to check that the rates of convergence are optimal in the minimax sense.

1.3 Related Works

Although it is attracting more and more interest, the use of persistent homology in data analysis remains widely heuristic. There are relatively few papers establishing connections between persistence and statistics and, despite a few promising results, the statistical analysis of homology, persistent homology and more general topological and geometric features of data is still in its infancy.

One of the first statistical results about persistent homology has been given in a parametric setting, by Bubenik and Kim (2007). They show for instance that for data sampled on an hypersphere according to a von-Mises Fisher distribution (among other distributions), the persistence diagrams of the density can be estimated with the parametric rate $n^{-1/2}$. However assuming that both the support and the parametric family of the distribution are known are strong assumptions which are hardly met in practice.

Closely related to our approach, statistical analysis of homology and of persistent homology has also been proposed very recently by Balakrishnan et al. (2012); Fasy et al. (2014) in the specific context of manifolds, i.e. when the geometric structure underlying the data is assumed to be a smooth submanifold of an Euclidean space. In the first paper, the authors exhibit minimax rates of convergence for the estimation of the Betti numbers of the underlying manifold under different models of noise. This approach is also strongly connected to manifold estimation results obtained by Genovese et al. (2012b). Related lower bounds have also been recently obtained by Weinberger (2014) in a different and more restrictive setting. Our results are in the same spirit as Balakrishnan et al. (2012) but extend to persistent homology and allow us to deal with general compact metric spaces. In the second paper, the authors develop several methods to find confidence sets for persistence diagrams using subsampling methods and kernel estimators among other approaches. Although they tackle a different problem, it has some connections with the problem considered in the present paper that we briefly mention in Section 3.4.

Both Fasy et al. (2014) and our work start from the observation that persistence diagram inference is strongly connected to the better known problem of support estimation. As far

as we know, only few results about support estimation in general metric spaces have been given in the past. An interesting framework is proposed by De Vito et al. (2014): in this paper the support estimation problem is tackled using kernel methods. On the other hand, a large amount of literature is available for measure support estimation in $\mathbb{R}^d$; see for instance the review by Cuevas (2009) for more details. Note that many results on this topic are given with respect to the volume of symmetric set difference (see for instance Biau et al., 2009, and references therein) while in our topological estimation setting we need convergence results for support estimation in Hausdorff metric.

The estimator $\widehat{\mathbb{X}}_n = \{X_1, \dots X_n\}$ and the Devroye and Wise (1980) estimator, $\hat{S}_n = \bigcup_{i=1}^n \bar{B}(X_i, \varepsilon_n)$, where $\bar{B}(x, \varepsilon)$ denotes the closed ball centered at x with radius ε, are both natural estimators of the support. The use of $\hat{S}_n$ is particularly relevant when the convergence of the measure of the symmetric set difference is considered but does not provide better results than $\widehat{\mathbb{X}}_n$ in our Hausdorff distance setting. The convergence rate of $\widehat{\mathbb{X}}_n$ to the support of the measure with respect to the Hausdorff distance is given by Cuevas and Rodríguez-Casal (2004) in $\mathbb{R}^d$. Support estimation in $\mathbb{R}^d$ has also been studied under various additional assumptions such as convexity assumptions (Dümbgen and Walther, 1996; Rodríguez-Casal, 2007; Cuevas et al., 2012) or through boundary fragments estimation (Korostelëv and Tsybakov, 1993; Korostelëv et al., 1995) just to name a few. Another classical assumption is that the measure has a density with respect to the Lebesgue measure. In this context, plug-in methods based on non parametric estimators of the density have been proposed by Cuevas and Fraiman (1997) and Tsybakov (1997). We consider persistence diagram estimation in the density framework of Singh et al. (2009) in Section 4 and show in this particular context that $\widehat{\mathbb{X}}_n$ allows us to define a persistence diagram estimator that reaches optimal rates of convergence in the minimax sense.

A few different methods have also been proposed for topology estimation in non-deterministic frameworks such as those based on deconvolution (Caillerie et al., 2011; Niyogi et al., 2011). Several recent attempts have also been made, with completely different approaches, to study persistence diagrams from a statistical point of view, such as Mileyko et al. (2011) who study probability measures on the space of persistence diagrams or Bubenik (2012) who introduces a functional representation of persistence diagrams, the so-called persistence landscapes, allowing means and variance of persistence diagrams to be defined. Notice that our results should easily extend to persistence landscapes.

The paper is organized as follows. Background notions and results on metric spaces, filtered simplicial complexes, and persistent homology that are necessary to follow the paper are presented in Section 2. The rates of convergence for the estimation of persistence diagrams in general metric spaces are established in Section 3. We also study these convergence rates in $\mathbb{R}^d$ for a few classical problems in Section 4. Some numerical experiments illustrating our results are given in Section 5. All the technical proofs are given in Appendix.

2. Background

We first recall the required background about measured metric spaces and persistent homology.

2.1 Metric Measure Spaces

Recall that a metric space is a pair $(\mathbb{M}, \rho)$ where $\mathbb{M}$ is a set and $\rho : \mathbb{M} \times \mathbb{M} \to \mathbb{R}$ is a nonnegative map such that for any $x, y, z \in \mathbb{M}$, $\rho(x, y) = 0$ if and only if $x = y$, $\rho(x, y) = \rho(y, x)$ and $\rho(x, z) \leqslant \rho(x, y) + \rho(y, z)$. We denote by $\mathcal{K}(\mathbb{M})$ the set of all the compact subsets of $\mathbb{M}$. For a point $x \in \mathbb{M}$ and a subset $C \in \mathcal{K}(\mathbb{M})$, the distance $d(x, C)$ of x to C is the minimum over all $y \in C$ of $d(x, y)$. The Hausdorff distance $\mathrm{d_H}(C_1, C_2)$ between two subsets $C_1, C_2 \in \mathcal{K}(\mathbb{M})$ is the maximum over all points in C_1 of their distance to C_2 and over all points in C_2 of their distance to C_1 :

$$\mathrm{d_H}(C_1, C_2) = \max\{ \sup_{x \in C_1} d(x, C_2),\ \sup_{y \in C_2} d(y, C_1) \}.$$

Note that $(\mathcal{K}(\mathbb{M}), \mathrm{d_H})$ is a metric space and can be endowed with its Borel σ-algebra.

Two compact metric spaces $(\mathbb{M}_1, \rho_1)$ and $(\mathbb{M}_2, \rho_2)$ are *isometric* if there exists a bijection $\Phi : \mathbb{M}_1 \to \mathbb{M}_2$ that preserves distances, namely: $\forall x, y \in \mathbb{M}_1$, $\rho_2(\Phi(x), \Phi(y)) = \rho_1(x, y)$. Such a map Φ is called an *isometry*. One way to compare two metric spaces is to measure how far these two metric spaces are from being isometric. The corresponding distance is called the *Gromov-Hausdorff distance* (see for instance Burago et al., 2001). Intuitively, it is the infimum of their Hausdorff distance over all possible isometric embeddings of these two spaces into a common metric space.

Definition 1 *Let $(\mathbb{M}_1, \rho_1)$ and $(\mathbb{M}_2, \rho_2)$ be two compact metric spaces. The Gromov-Hausdorff distance $\mathrm{d}_{GH}((\mathbb{M}_1, \rho_1), (\mathbb{M}_2, \rho_2))$ is the infimum of the real numbers $r \geqslant 0$ such that there exist a metric space $(\mathbb{M}, \rho)$ and subspaces C_1 and C_2 in $\mathcal{K}(\mathbb{M})$ which are isometric to $\mathbb{M}_1$ and $\mathbb{M}_2$ respectively and such that $\mathrm{d}_H(C_1, C_2) < r$. The Gromov-Hausdorff distance d_{GH} defines a metric on the space $\mathcal{K}$ of isometry classes of compact metric spaces (see Burago et al., 2001, Theorem 7.3.30).*

Notice that when $\mathbb{M}_1$ and $\mathbb{M}_2$ are subspaces of a same metric space $(\mathbb{M}, \rho)$ then $\mathrm{d_{GH}}(\mathbb{M}_1, \mathbb{M}_2) \leqslant \mathrm{d_H}(\mathbb{M}_1, \mathbb{M}_2)$.

2.2 Measure

Let μ be a probability measure on $(\mathbb{M}, \rho)$ equipped with its Borel algebra. Let $\mathbb{X}_\mu$ denote the support of the measure μ, namely the smallest closed set with probability one. In the following of the paper, we will assume that $\mathbb{X}_\mu$ is compact and thus $\mathbb{X}_\mu \in \mathcal{K}(\mathbb{M})$. Also note that $(\mathbb{X}_\mu, \rho) \in \mathcal{K}$.

The main assumption we will need in the following of the paper provides a lower bound on the measure μ. We say that μ satisfies the *standard assumption* if there exist $a' > 0$, $r_0 > 0$ and $b > 0$ such that

$$\forall x \in \mathbb{X}_\mu,\ \forall r \in (0, r_0),\ \mu(B(x, r)) \geqslant a' r^b \tag{2.1}$$

where $B(x, r)$ denotes the open ball of center x and radius r in $\mathbb{M}$. This assumption is popular in the literature about set estimation (see for instance Cuevas, 2009) but it has generally been considered with $b = d$ in $\mathbb{R}^d$. Since $\mathbb{X}_\mu$ is compact, reducing the constant a' to a smaller constant a if necessary, we easily check that assumption (2.1) is equivalent to

$$\forall x \in \mathbb{X}_\mu,\ \forall r > 0,\ \mu(B(x, r)) \geqslant 1 \wedge a r^b \tag{2.2}$$

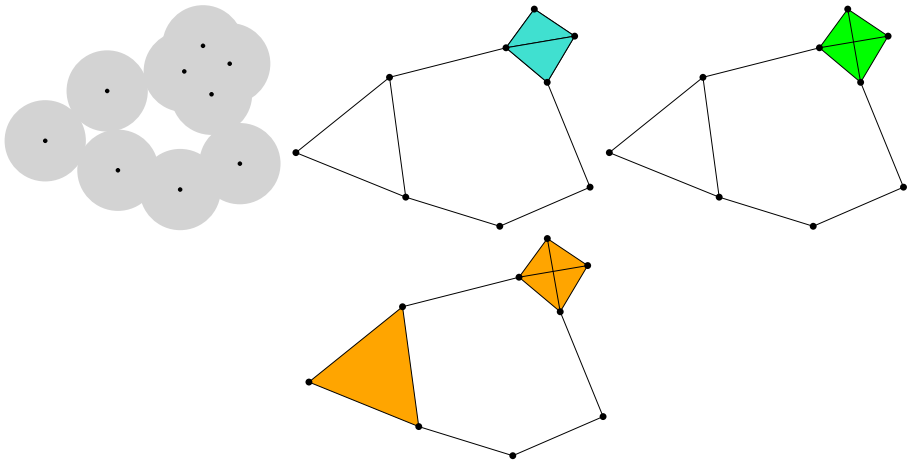


Figure 2: From left to right: the α sublevelset of the distance function to a point set $\mathbb{X}$ in $\mathbb{R}^2$, the α-complex, $\text{Cech}_\alpha(\mathbb{X})$ and $\text{Rips}_{2\alpha}(\mathbb{X})$. The last two include a tetrahedron.

where $x \wedge y$ denotes the minimum between x and y. We then say that μ satisfies the (a, b)-*standard assumption*.

2.3 Simplicial Complexes on Metric Spaces

The geometric complexes we consider in this paper are built on top of metric spaces and come as nested families indexed by a real parameter. Topological persistence is used to infer and encode the evolution of the topology of theses families as the parameter grows. For a complete definition of these geometric filtered complexes built on top of metric spaces and their use in TDA, we refer to Chazal et al. (2012b), Section 4.2. Here we only give a brief reminder and refer to Figure 2 for illustrations. A simplicial complex $\mathcal{C}$ is a set of simplexes (points, segments, triangles, etc) such that any face from a simplex in $\mathcal{C}$ is also in $\mathcal{C}$ and the intersection of any two simplices of $\mathcal{C}$ is a (possibly empty) face of these simplices. Notice that we do not assume such simplicial complexes to be finite. The complexes we consider in this paper can be seen as a generalization of neighborhood graphs in dimension larger than 1.

Given a metric space $\mathbb{X}$ which will also serve as the vertex set, the *Vietoris-Rips complex* $\text{Rips}_\alpha(\mathbb{X})$ is the set of simplices $[x_0, \ldots, x_k]$ such that $d_{\mathbb{X}}(x_i, x_j) \leqslant \alpha$ for all (i, j). The *Čech complex* $\text{Cech}_\alpha(\mathbb{X})$ is similarly defined as the set of simplices $[x_0, \ldots, x_k]$ such that the $k+1$ closed balls $B(x_i, \alpha)$ have a non-empty intersection. Note that these two complexes are related by $\text{Rips}_\alpha(\mathbb{X}) \subseteq \text{Cech}_\alpha(\mathbb{X}) \subseteq \text{Rips}_{2\alpha}(\mathbb{X})$. Note also that these two families of complexes only depend on the pairwise distances between the points of $\mathbb{X}$.

When $\mathbb{X}$ is embedded in some larger metric space $\mathbb{M}$, we can extend the definition of the Čech complex to the set of simplices $[x_0, \ldots, x_k]$ such that the $k+1$ closed balls $B(x_i, \alpha)$ have a non-empty intersection in $\mathbb{M}$ (not just in $\mathbb{X}$). We can also define the *alpha-complex* or α-*complex* as the set of simplices $[x_0, \ldots, x_k]$ such that, for some $\beta \leqslant \alpha$ that depends on the simplex, the $k+1$ closed balls $B(x_i, \beta)$ and the complement of all the other balls $B(x, \beta)$ for

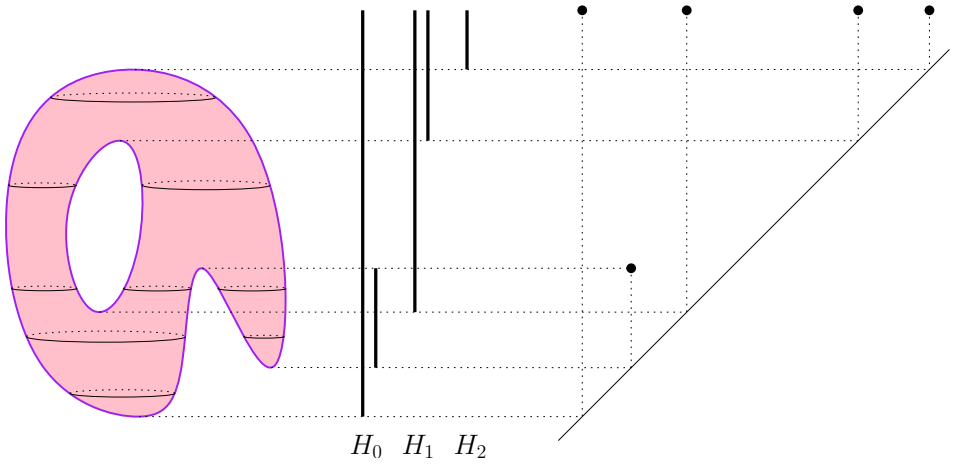


Figure 3: A torus $\mathbb{T}$ filtered by its z-coordinate: $\text{Filt}_\alpha = \{P \in \mathbb{T} | P_z \leqslant \alpha\}$, its persistence barcode, and its persistence diagram.

$x \in \mathbb{X}$ have a non-empty intersection in $\mathbb{M}$. In the particular case where $\mathbb{M} = \mathbb{R}^d$, those two complexes have the same homotopy type (they are equivalent for our purposes) as the union of the balls $B(x, \alpha)$ for $x \in \mathbb{X}$, as in Figure 2, and the α-complex only contains simplices of dimension at most d. Note that the union of the balls $B(x, \alpha)$ is also the α-sublevel set of the distance to $\mathbb{X}$ function $d(\cdot, \mathbb{X})$, and as a consequence, those filtrations thus provide a convenient way to study the evolution of the topology of the union of growing balls or the sublevel sets of $d(\cdot, \mathbb{X})$ (see Figure 2 and Section 5 for more examples).

There are several other families that we could also have considered, most notably witness complexes (Chazal et al., 2012b). Extending our results to them is straightforward and yields very similar results, so we will restrict to the families defined above in the rest of the paper.

All these families of complexes have the fundamental property that they are non-decreasing with α; for any $\alpha \leqslant \beta$, there is an inclusion of $\text{Rips}_\alpha(\mathbb{X})$ in $\text{Rips}_\beta(\mathbb{X})$, and similarly for the Čech, and Alpha complexes. They are thus called *filtrations*. In the following, the notation $\text{Filt}(\mathbb{X}) := (\text{Filt}_\alpha(\mathbb{X}))_{\alpha \in \mathcal{A}}$ denotes one of the filtrations defined above.

2.4 Persistence Diagrams

An extensive presentation of persistence diagrams is available in Chazal et al. (2012a). We recall a few definitions and results that are needed in this paper.

We first give the intuition behind persistence. Given a filtration as above, the topology of $\text{Filt}_\alpha(\mathbb{X})$ changes as α increases: new connected components can appear, existing connected components can merge, cycles and cavities can appear and can be filled, etc. Persistent homology is a tool that tracks these changes, identifies *features* and associates a *lifetime* to them. For instance, a connected component is a feature that is born at the smallest α such that the component is present in $\text{Filt}_\alpha(\mathbb{X})$, and dies when it merges with an older connected component. Intuitively, the longer a feature persists, the more relevant it is.

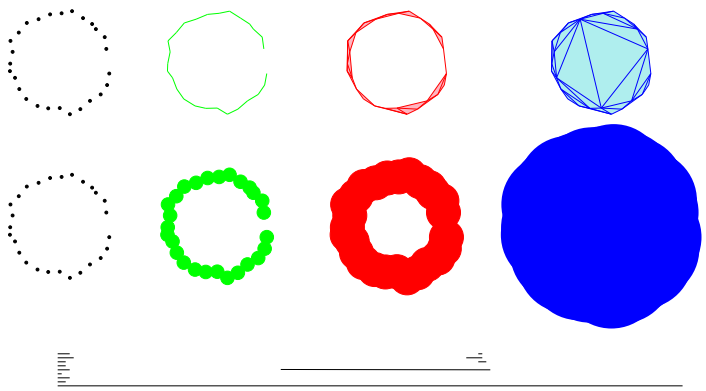


Figure 4: An α-complex filtration, the sublevelset filtration of the distance function, and their common persistence barcode (they are homotopy equivalent).

We now formalize the presentation a bit. Given a filtration as above, we can consider the $\mathbb{Z}_2$-homology groups [1] of the simplicial complexes and get a sequence of vector spaces $(H(\mathrm{Filt}_\alpha(\mathbb{X})))_{\alpha\in\mathcal{A}}$, where the inclusions $\mathrm{Filt}_\alpha(\mathbb{X}) \subseteq \mathrm{Filt}_\beta(\mathbb{X})$ induce linear maps $H(\mathrm{Filt}_\alpha(\mathbb{X})) \to H(\mathrm{Filt}_\beta(\mathbb{X}))$. In many cases, this sequence can be decomposed as a direct sum of intervals, where an interval is a sequence of the form

$$0 \to \ldots \to 0 \to \mathbb{Z}_2 \to \ldots \to \mathbb{Z}_2 \to 0 \to \ldots \to 0$$

(the linear maps $\mathbb{Z}_2 \to \mathbb{Z}_2$ are all the identity). These intervals can be interpreted as features of the (filtered) complex, such as a connected component or a loop, that appear at parameter α_{birth} in the filtration and disappear at parameter α_{death}. An interval is determined uniquely by these two parameters. It can be represented as a segment whose extremities have abscissae α_{birth} and α_{death}; the set of these segments is called the barcode of $\mathrm{Filt}(\mathbb{X})$. An interval can also be represented as a point in the plane, where the x-coordinate indicates the birth time and the y-coordinate the death time. The set of points (with multiplicity) representing the intervals is called the persistence diagram $\mathsf{dgm}(\mathrm{Filt}(\mathbb{X}))$. Note that the diagram is entirely contained in the half-plane above the diagonal Δ defined by $y = x$, since death always occurs after birth. Chazal et al. (2012a) show that this diagram is still well-defined even in cases where the sequence might not be decomposable as a finite sum of intervals, and in particular $\mathsf{dgm}(\mathrm{Filt}(\mathbb{X}))$ is well-defined for any compact metric space $\mathbb{X}$ (Chazal et al., 2012b). Note that for technical reasons, the points of the diagonal Δ are considered as part of every persistence diagram, with infinite multiplicity. The most persistent features (supposedly the most important) are those represented by the longest bars in the barcode, i.e. the points furthest from the diagonal in the diagram, whereas points close to the diagonal can be interpreted as noise.

1. The notion of (simplicial) homology is a classical concept in algebraic topology that provides powerful tools to formalize and handle the notion of topological features of a simplicial complex in an algebraic way. For example the 0-dimensional homology group H_0 represents the 0-dimensional features, i.e. the connected components of the complex, H_1 represents the 1-dimensional features (cycles), H_2 represents the 2-dimensional features (cavities),... See Hatcher (2001) for an introduction to simplicial homology.

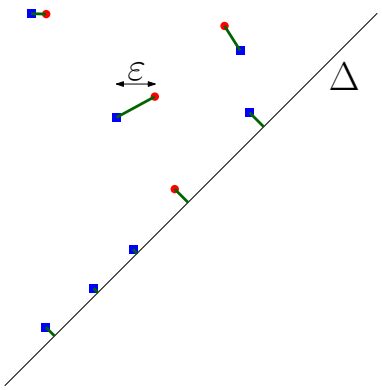


Figure 5: Two diagrams at bottleneck distance ε.

The space of persistence diagrams is endowed with a metric called the *bottleneck distance* $\mathrm{d_b}$. Given two persistence diagrams, it is defined as the infimum, over all perfect matchings of their points, of the largest L^∞-distance between two matched points, see Figure 5. The presence of the diagonal in all diagrams means we can consider partial matchings of the off-diagonal points, and the remaining points are matched to the diagonal. With more details, given two diagrams dgm_1 and dgm_2, we can define a matching m as a subset of $\mathsf{dgm}_1 \times \mathsf{dgm}_2$ such that every point of $\mathsf{dgm}_1 \backslash \Delta$ and $\mathsf{dgm}_2 \backslash \Delta$ appears exactly once in m. The bottleneck distance is then:

$$\mathrm{d_b}(\mathsf{dgm}_1, \mathsf{dgm}_2) = \inf_{\text{matching } m} \max_{(p,q)\in m} ||q-p||_\infty.$$

Note that points close to the diagonal Δ are easily matched to the diagonal, which fits with their interpretation as irrelevant noise.

A fundamental property of persistence diagrams, proved by Chazal et al. (2012a), is their *stability*. If $\mathbb{X}$ and $\tilde{\mathbb{X}}$ are two compact metric spaces then one has

$$\mathrm{d_b}\left(\mathsf{dgm}(\mathrm{Filt}(\mathbb{X})), \mathsf{dgm}(\mathrm{Filt}(\tilde{\mathbb{X}}))\right) \leqslant 2\mathrm{d_{GH}}\left(\mathbb{X}, \tilde{\mathbb{X}}\right). \tag{2.3}$$

Moreover, if $\mathbb{X}$ and $\tilde{\mathbb{X}}$ are embedded in the same metric space $(\mathbb{M}, \rho)$ then one has

$$\mathrm{d_b}\left(\mathsf{dgm}(\mathrm{Filt}(\mathbb{X})), \mathsf{dgm}(\mathrm{Filt}(\tilde{\mathbb{X}}))\right) \leqslant 2\mathrm{d_{GH}}\left(\mathbb{X}, \tilde{\mathbb{X}}\right) \leqslant 2\mathrm{d_H}\left(\mathbb{X}, \tilde{\mathbb{X}}\right). \tag{2.4}$$

Notice that these properties are only metric properties: they do not involve here any probability measure on $\mathbb{X}$ and $\tilde{\mathbb{X}}$.

3. Persistence Diagram Estimation in Metric Spaces

Let $(\mathbb{M}, \rho)$ be a metric space. Assume that we observe n points $X_1 \ldots, X_n$ in $\mathbb{M}$ drawn i.i.d. from some unknown measure μ whose support is a compact set denoted $\mathbb{X}_\mu$.

3.1 From Support Estimation to Persistence Diagram Estimation

The Gromov-Hausdorff distance allows us to compare $\mathbb{X}_\mu$ with compact metric spaces not necessarily embedded in $\mathbb{M}$. We thus consider $(\mathbb{X}_\mu, \rho)$ as an element of $\mathcal{K}$ (rather than an

element of $\mathcal{K}(\mathbb{M})$). In the following, an *estimator* $\widehat{\mathbb{X}}$ of $\mathbb{X}_\mu$ is a function of $X_1 \ldots, X_n$ that takes values in $\mathcal{K}$ and which is measurable for the Borel algebra induced by $\mathrm{d_{GH}}$.

Let $\mathrm{Filt}(\mathbb{X}_\mu)$ and $\mathrm{Filt}(\widehat{\mathbb{X}})$ be filtrations defined on $\mathbb{X}_\mu$ and $\widehat{\mathbb{X}}$. The statistical analysis of persistence diagrams proposed above starts from the following key fact: according to (2.3), for any $\varepsilon > 0$:

$$\mathbb{P}\left(\mathrm{d_b}\left(\mathsf{dgm}(\mathrm{Filt}(\mathbb{X}_\mu)), \mathsf{dgm}(\mathrm{Filt}(\widehat{\mathbb{X}}))\right) > \varepsilon\right) \leqslant \mathbb{P}\left(\mathrm{d_{GH}}(\mathbb{X}_\mu, \widehat{\mathbb{X}}) > 2\varepsilon\right) \tag{3.1}$$

where the probability corresponds to the product measure $\mu^{\otimes n}$. Our strategy then consists in finding an estimator of the support which is close to $\mathbb{X}_\mu$ for the $\mathrm{d_{GH}}$ distance. Note that this general strategy of estimating $\mathbb{X}_\mu$ in $\mathcal{K}$ is not only of theoretical interest. Indeed, as mentioned in the introduction, in some cases the space $\mathbb{M}$ is unknown and the observations $X_1 \ldots, X_n$ are just known through their matrix of pairwise distances $\rho(X_i, X_j)$, $i, j = 1, \cdots, n$. The use of the Gromov-Hausdorff distance then allows us to consider this set of observations as an abstract metric space of cardinality n independently of the way it is embedded in $\mathbb{M}$.

This general framework includes the more standard approach consisting in estimating the support by restraining the values of $\widehat{\mathbb{X}}$ to $\mathcal{K}(\mathbb{M})$. According to (2.4), in this case, for any $\varepsilon > 0$:

$$\mathbb{P}\left(\mathrm{d_b}\left(\mathsf{dgm}(\mathrm{Filt}(\mathbb{X}_\mu)), \mathsf{dgm}(\mathrm{Filt}(\widehat{\mathbb{X}}))\right) > \varepsilon\right) \leqslant \mathbb{P}\left(\mathrm{d_H}(\mathbb{X}_\mu, \widehat{\mathbb{X}}) > 2\varepsilon\right). \tag{3.2}$$

Using equations (3.1) and (3.2) the problem of persistence diagrams estimation reduces to the better known problem of estimating the support of a measure.

Let $\widehat{\mathbb{X}}_n := \{X_1, \ldots, X_n\}$ be a set of independent observations sampled according to μ endowed with the restriction of the distance ρ to this set. This finite metric space is a natural estimator of the support $\mathbb{X}_\mu$. In several contexts discussed in the following, $\widehat{\mathbb{X}}_n$ shows optimal rates of convergence for the estimation of $\mathbb{X}_\mu$ with respect to the Hausdorff and Gromov-Hausdorff distance. From (3.2) we will then obtain upper bounds on the rate of convergence of $\mathrm{Filt}(\widehat{\mathbb{X}}_n)$. We also obtain the corresponding lower bounds to prove optimality.

In the next subsection, we tackle persistence diagram estimation in the general framework of abstract metric spaces. We will consider more particular contexts later in the paper.

3.2 Convergence of Persistence Diagrams

Cuevas and Rodríguez-Casal (2004) give the rate of convergence in Hausdorff distance of $\widehat{\mathbb{X}}_n$ for some probability measure μ satisfying an (a, b)-standard assumption on $\mathbb{R}^d$. In this section, we consider the more general context where μ is a probability measure satisfying an (a, b)-standard assumption on a metric space $(\mathbb{M}, \rho)$, with $b > 0$. We give below the rate of convergence of $\widehat{\mathbb{X}}_n$ in this context. The proof follows the lines of the proof of Cuevas and Rodríguez-Casal (2004, Theorem 3).

Theorem 2 *Assume that a probability measure μ on $\mathbb{M}$ satisfies the (a, b)-standard assumption. Then, for any $\varepsilon > 0$:*

$$\mathbb{P}\left(\mathrm{d}_H(\mathbb{X}_\mu, \widehat{\mathbb{X}}_n) > 2\varepsilon\right) \leqslant \frac{2^b}{a\varepsilon^b} \exp(-na\varepsilon^b) \wedge 1.$$

Moreover, there exist two constants C_1 and C_2 only depending on a and b such that

$$\limsup_{n\to\infty} \left(\frac{n}{\log n}\right)^{1/b} \mathrm{d}_H(\mathbb{X}_\mu, \widehat{\mathbb{X}}_n) \leqslant C_1 \qquad \textit{almost surely,}$$

and

$$\lim_{n\to\infty} \mathbb{P}\left(\mathrm{d}_H(\mathbb{X}_\mu, \widehat{\mathbb{X}}_n) \leqslant C_2 \left(\frac{\log n}{n}\right)^{1/b}\right) = 1.$$

Since $\mathrm{d}_{\mathrm{GH}}(\mathbb{X}_\mu, \widehat{\mathbb{X}}_n) \leqslant \mathrm{d}_{\mathrm{H}}(\mathbb{X}_\mu, \widehat{\mathbb{X}}_n)$ the above theorem also holds when the Gromov distance is replaced by the Gromov-Hausdorff distance. In practice this allows to consider $\widehat{\mathbb{X}}_n$ as an abstract metric space without taking care of the way it is embedded in the, possibly unknown, metric space $\mathbb{M}$.

Using (3.1) and (2.4), we then derive from the previous result the following corollary for the convergence rate of the persistence diagram $\mathrm{Filt}(\widehat{\mathbb{X}}_n)$ toward $\mathrm{Filt}(\mathbb{X}_\mu)$.

Corollary 3 *Assume that the probability measure μ on $\mathbb{M}$ satisfies the (a,b)-standard assumption, then for any $\varepsilon > 0$:*

$$\mathbb{P}\left(\mathrm{d_b}\left(\mathsf{dgm}(\mathrm{Filt}(\mathbb{X}_\mu)), \mathsf{dgm}(\mathrm{Filt}(\widehat{\mathbb{X}}_n))\right) > \varepsilon\right) \leqslant \frac{2^b}{a\varepsilon^b} \exp(-na\varepsilon^b) \wedge 1. \tag{3.3}$$

Moreover,

$$\limsup_{n\to\infty} \left(\frac{n}{\log n}\right)^{1/b} \mathrm{d_b}\left(\mathsf{dgm}(\mathrm{Filt}(\mathbb{X}_\mu)), \mathsf{dgm}(\mathrm{Filt}(\widehat{\mathbb{X}}_n))\right) \leqslant C_1 \qquad \textit{almost surely,}$$

and

$$\lim_{n\to\infty} \mathbb{P}\left(\mathrm{d_b}\left(\mathsf{dgm}(\mathrm{Filt}(\mathbb{X}_\mu)), \mathsf{dgm}(\mathrm{Filt}(\widehat{\mathbb{X}}_n))\right) \leqslant C_2 \left(\frac{\log n}{n}\right)^{1/b}\right) = 1$$

where C_1 and C_2 are the same constants as in Theorem 2.

3.3 Minimax Optimal Rate of Convergence

Let $\mathcal{P}(a,b,\mathbb{M})$ be the set of all the probability measures on the metric space $(\mathbb{M},\rho)$ satisfying the (a,b)-standard assumption on $\mathbb{M}$:

$$\mathcal{P}(a,b,\mathbb{M}) := \left\{\mu \text{ on } \mathbb{M} \mid \mathbb{X}_\mu \text{ is compact and } \forall x \in \mathbb{X}_\mu,\ \forall r > 0,\ \mu\left(B(x,r)\right) \geqslant 1 \wedge ar^b\right\}.$$

The next theorem gives upper and lower bounds for the rate of convergence of persistence diagrams. The upper bound comes as a consequence of Corollary 3, while the lower bound is established using the so-called Le Cam's lemma (see Lemma 9 in Appendix).

Theorem 4 *Let $(\mathbb{M},\rho)$ be a metric space and let $a > 0$ and $b > 0$. Then:*

$$\sup_{\mu\in\mathcal{P}(a,b,\mathbb{M})} \mathbb{E}\left[\mathrm{d_b}(\mathsf{dgm}(\mathrm{Filt}(\mathbb{X}_\mu)), \mathsf{dgm}(\mathrm{Filt}(\widehat{\mathbb{X}}_n)))\right] \leqslant C\left(\frac{\log n}{n}\right)^{1/b} \tag{3.4}$$

where the constant C only depends on a and b (not on $\mathbb{M}$). Assume moreover that there exists a non isolated point x in $\mathbb{M}$ and consider any sequence $(x_n) \in (\mathbb{M}\backslash\{x\})^{\mathbb{N}}$ such that $\rho(x, x_n) \leqslant (an)^{-1/b}$. Then for any estimator $\widehat{\mathsf{dgm}}_n$ of $\mathsf{dgm}(\mathrm{Filt}(\mathbb{X}_\mu))$:

$$\liminf_{n\to\infty} \rho(x, x_n)^{-1} \sup_{\mu\in\mathcal{P}(a,b,\mathbb{M})} \mathbb{E}\left[\mathrm{d_b}(\mathsf{dgm}(\mathrm{Filt}(\mathbb{X}_\mu)), \widehat{\mathsf{dgm}}_n)\right] \geqslant C'$$

where C' is an absolute constant.

Consequently, the estimator $\mathsf{dgm}(\mathrm{Filt}(\hat{\mathbb{X}}_n))$ is minimax optimal on the space $\mathcal{P}(a, b, \mathbb{M})$ up to a logarithmic term as soon as we can find a non-isolated point in $\mathbb{M}$ and a sequence (x_n) in $\mathbb{M}$ such that $\rho(x_n, x) \sim (an)^{-1/b}$. This is obviously the case for the Euclidean space $\mathbb{R}^d$.

One classical method to obtain tight lower bounds with sup norm metrics is applying a Fano's strategy based on several hypotheses (see for instance Tsybakov and Zaiats, 2009, Chapter 2). Applying this method is more difficult than it seems in our context. Indeed, the bottleneck distance makes tricky the construction of multiple hypotheses. However, in specific cases, we can obtain the matching lower bound with a more direct proof.

Theorem 5 *Consider $(\frac{1}{2}, 1)$-standard measures on the unit segment $[0, 1]$. For any estimator $\widehat{\mathsf{dgm}}_n$ of $\mathsf{dgm}(\mathrm{Filt}(\mathbb{X}_\mu))$:*

$$\liminf_{n\to\infty} \sup_{\mu\in\mathcal{P}(\frac{1}{2},1,[0,1])} \frac{n}{\log n}\mathbb{E}\left[\mathrm{d_b}(\mathsf{dgm}(\mathrm{Filt}(\mathbb{X}_\mu)), \widehat{\mathsf{dgm}}_n)\right] \geqslant C$$

where C is an absolute constant.

It should be straightforward to extend this to measures on the cube $[0, 1]^b$, as long as b is an integer, with a lower-bound of $C_b(\frac{\log n}{n})^{1/b}$. Note that this bound applies to the homology of dimension b. It is possible that lower-dimensional homology may be easier to estimate.

3.4 Confidence Sets for Persistence Diagrams

Corollary 3 can also be used to find confidence sets for persistence diagrams. Assume that a and b are known and let $\Psi : \eta \to \exp(-\eta)/\eta$. Then for $\alpha \in (0, 1)$,

$$B_{d_b}\left(\mathsf{dgm}(\mathrm{Filt}(\mathbb{X}_\mu)), \left[\frac{1}{na}\Psi^{-1}\left(\frac{\alpha}{n2^b}\right)\right]^{1/b}\right)$$

is a confidence region for $\mathsf{dgm}\left(\mathrm{Rips}(\mu(K))\right)$ of level $1 - \alpha$. Nevertheless, in practice the coefficients a and b can be unknown. In $\mathbb{R}^d$, the coefficient b can be taken equal to the ambient dimension d in many situations. Finding lower bounds on the coefficient a is a tricky problem that is out of the scope of the paper. Alternative solutions have been proposed recently by Fasy et al. (2014) and we refer the reader to this paper for more details.

4. Persistence Diagram Estimation in $\mathbb{R}^k$

In this section, we study the convergence rates of persistence diagram estimators for data embedded in $\mathbb{R}^k$. In particular we study two situations of interest proposed respectively by Singh et al. (2009) and Genovese et al. (2012b) in the context of measure support estimation. In the first situation the measure has a density with respect to the Lebesgue measure on $\mathbb{R}^d$ whose behavior is controlled near the boundary of its support. In the second case, the measure is supported on a manifold. These two frameworks are complementary and provide realistic frameworks for topological inference in $\mathbb{R}^d$.

4.1 Minimax Optimal Persistence Diagram Estimation for Nonsingular Measures on $\mathbb{R}^k$

The paper by Singh et al. (2009) is a significant breakthrough for level set estimation through density estimation. It presents a fully data-driven procedure, in the spirit of Lepski's method, that is adaptive to unknown local density regularity and achieves a Hausdorff error control that is minimax optimal for a class of level sets with very general shapes. In particular, the assumptions of Singh et al. (2009) describe the smoothness of the density near the boundary of the support.

In this section, we propose to study persistence diagram inference in the framework of Singh et al. (2009) since this framework is very intuitive and natural. Nevertheless, we do not use the estimator of Singh et al. (2009) for this task since we only consider here the support estimation problem (and not the more general level set issue as in Singh et al., 2009). Indeed, we will see that the estimator $\hat{\mathbb{X}}_n$ has the optimal rate of convergence for estimating the support according to $\mathrm{d_H}$, as well as for estimating the persistence diagram. We now recall the framework of Singh et al. (2009, Section 4.3) corresponding to support set estimation.

Let $X_1, \dots, X_n$ be i.i.d. observations drawn from an unknown probability measure μ having density f with respect to the Lebesgue measure and defined on a compact set $\chi \subset \mathbb{R}^k$. Let $\mathbb{X}_f$ denote the support of μ, and let $G_0 := \{x \in \chi \mid f(x) > 0\}$. The boundary of a set G is denoted ∂G and for any $\varepsilon > 0$, $I_\varepsilon(G) := \bigcup_{x \mid B(x,\varepsilon) \subset G} B(x, \varepsilon)$ is the ε-inner of G. The two main assumptions of Singh et al. (2009) are the following:

[A]: The density f is upper bounded by $f_{\max} > 0$ and there exist constants α, C_a, $\delta_a > 0$ such that for all $x \in G_0$ with $f(x) \leqslant \delta_a$, $f(x) \geqslant C_a\, d(x, \partial G_0)^\alpha$.

[B]: There exist constants $\varepsilon_0 > 0$ and $C_b > 0$ such that for all $\varepsilon \leqslant \varepsilon_0$, $I_\varepsilon(G_0) \neq \varnothing$ and $d(x, I_\varepsilon(G_0)) \leqslant C_b\, \varepsilon$ for all $x \in \partial G_0$.

We denote by $\mathcal{F}(\alpha)$ the set composed of all the densities on χ satisfying assumptions [A] and [B], for a fixed set of positive constants C_a, C_b, δ_a, ε_0, $f_{\max}$, p and α.

Assumption [A] describes how fast the density increases in the neighborhood of the boundary of the support: the smaller α, the easier the support may be possible to detect. Assumption [B] prevents the boundary from having arbitrarily small features (as for cusps). We refer to Singh et al. (2009) for more details and discussions about these two assumptions and their connections with assumptions in other works.

For persistence diagram estimation, we are interested in estimating the support $\mathbb{X}_f$ whereas the assumptions [A] and [B] involve the set G_0. However, as stated in Lemma 11

(given in Appendix B.4), these two sets are here almost identical in the sense that $\mathrm{d_H}(G_0, \mathbb{X}_f) = 0$. Moreover, it can be proved that under assumptions $[A]$ and $[B]$, the measure μ also satisfies the standard assumption with $b = \alpha + k$ (see Lemma 11). According to Proposition 4, the estimator $\mathsf{dgm}(\mathrm{Filt}(\widehat{\mathbb{X}}_n))$ thus converges in expectation to $\mathsf{dgm}(\mathrm{Filt}(\mathbb{X}_f))$ for d_b with a rate upper bounded by $(\log n/n)^{1/(k+\alpha)}$. We also show that this rate is minimax over the sets $\mathcal{F}(\alpha)$ by adapting the ideas of the proof given by Singh et al. (2009) for the Hausdorff lower bound.

Proposition 6 *1. For all $n \geqslant 1$,*

$$sup_{f\in\mathcal{F}(\alpha)}\mathbb{E}\left[\mathrm{d_b}(\mathsf{dgm}(\mathrm{Filt}(\mathbb{X}_f)), \mathsf{dgm}(\mathrm{Filt}(\widehat{\mathbb{X}}_n))\right] \leqslant C\left(\frac{n}{\log n}\right)^{-1/(k+\alpha)}$$

where C is a constant depending only on C_a, C_b, δ_a, ε_0, f_{max}, p and α.

2. There exists $c > 0$ such that

$$\inf_{\widehat{\mathsf{dgm}}_n}\sup_{f\in\mathcal{F}(\alpha)}\mathbb{E}\left[\mathrm{d_b}(\mathsf{dgm}(\mathrm{Filt}(\mathbb{X}_f)), \widehat{\mathsf{dgm}}_n)\right] \geqslant cn^{-1/(k+\alpha)}$$

for n large enough. The infimum is taken over all possible estimators $\widehat{\mathsf{dgm}}_n$ of $\mathsf{dgm}(\mathrm{Filt}(\mathbb{X}_f))$ based on n observations.

Remark 7 *The paper by Singh et al. (2009) is more generally about adaptive level set estimation. For this problem, Singh et al. define an histogram based estimator. Let $\mathcal{A}_j$ denote the collection of cells, in a regular partition of $\chi = [0,1]^k$ into hypercubes of dyadic side length 2^{-j}. Their estimator $\hat{f}$ is the histogram $\hat{f}(A) = \hat{P}(A)/\mu(A)$, where $\hat{P}(A) = \sum_{i=1\ldots n} 1_{X_i\in A}$. For estimating the level set $G_\gamma := \{x|f(x) \geqslant \gamma\}$, they consider the estimator*

$$\hat{G}_{\gamma,j} = \bigcup_{A\in\mathcal{A}_j \mid \hat{f}(A)>\gamma} A.$$

It is proved by Singh et al. (2009) that $\hat{G}_{\gamma,\hat{j}}$ achieves optimal rates of convergence for estimating the level sets, with $\hat{j}$ chosen in a data driven way. Concerning support estimation, they also show that $\hat{G}_{0,j}$ achieves optimal rates of convergence for estimating G_0. We have seen that in this context it is also the case for the estimator $\mathbb{X}_n$. Since no knowledge of α is required for this last estimator, we thus prefer to use this simpler estimator in this context.

4.2 Minimax Optimal Rates of Convergence of Persistence Diagram Estimation for Singular Measures in $\mathbb{R}^D$

We now consider the case where the support of μ is a smooth submanifold of $\mathbb{R}^D$. As far as we know, rates of convergence for manifold estimation, namely for the estimation of the support of a singular probability measure supported on a Riemannian manifold of $\mathbb{R}^D$, have only been studied recently by Genovese et al. (2012b,a). These papers assume several noise models, which all could be considered in the context of persistence diagram estimation. However, for the sake of simplicity, we only study here the problem where no additional noise is observed, which is referred as the *noiseless model* in the first of these two papers.

As before, upper bounds given by Genovese et al. (2012b) on the rates of convergence for the support estimation in Hausdorff distance directly provide upper bounds on the rates of convergence of the persistence diagram of the support. Before giving the rates of convergence we first recall and discuss the assumptions of Genovese et al. (2012b).

For any $r > 0$ and any set $A \subset \mathbb{R}^D$, let $A \oplus \varepsilon := \bigcup_{a \in A} B(a, r)$. Let $\Delta(\mathbb{X}_\mu)$ be the largest r such that each point in $\mathbb{X}_\mu \oplus r$ has a unique projection onto $\mathbb{X}_\mu$, this quantity has been introduced by Federer (1959), it is called reach or condition number in the literature.

For a fixed positive integer $k < D$, for some fixed positive constants b, B, κ and for a fixed compact domain χ in $\mathbb{R}^D$, Genovese et al. (2012b) define the set of probability measures $\mathcal{H} := \mathcal{H}(d, A, B, \kappa, \chi)$ on χ satisfying the two following assumptions:

- $[H_1]$ The support of the measure μ is a compact Riemannian manifold $\mathbb{X}_\mu$ (included in χ) of dimension k whose reach satisfies

$$\Delta(\mathbb{X}_\mu) \geqslant \kappa. \tag{4.1}$$

- $[H_2]$ The measure μ is assumed to have a density g with respect to k-dimensional volume measure vol_k on $\mathbb{X}_\mu$, such that

$$0 < A \leqslant \inf_{y \in \mathbb{X}_\mu} g(y) \leqslant \sup_{y \in \mathbb{X}_\mu} g(y) \leqslant B < \infty. \tag{4.2}$$

These two assumptions can be easily connected to the standard assumption. Indeed, according to Niyogi et al. (2008) and using $[H_1]$, for all $r \leqslant \kappa$ there exists some constant $C > 0$ such that for any $x \in \mathbb{X}_\mu$, we have

$$\begin{aligned} vol_k\left(B(x, r) \cap \mathbb{X}_\mu\right) &\geqslant C\left(1 - \frac{r^2}{4\kappa^2}\right)^{k/2} r^k \\ &\geqslant C' r^k \end{aligned}$$

and the same holds for μ according to $[H_2]$. Under these two assumptions, μ satisfies the standard assumption with $b = k$. Thus, if we take $\hat{\mathbb{X}}_n$ for estimating the support $\mathbb{X}_\mu$ in this context, we obtain a rate of convergence upper bounded by $(\frac{\log n}{n})^{1/k}$ both for support and persistence diagram estimation. Nevertheless, this rate is not minimax optimal for estimating the support on the spaces $\mathcal{H}$ as shown by Genovese et al. (2012b, Theorem 2). The correct minimax rate is $n^{-2/k}$ and the same is true for the persistence diagram estimation, as stated in the following proposition. However, the achievement of this optimal rate relies on a "theoretical" estimator proposed by Genovese et al. (2012b) that can not be computed in practice.

Proposition 8 *Assume that we observe an n-sample under the previous assumptions, then there exist two constants C and C' depending only on $\mathcal{H}$ such that*

$$Cn^{-2/k} \leqslant \inf_{\widehat{\mathsf{dgm}}_n} \sup_{\mu \in \mathcal{H}} \mathbb{E}\left[\mathrm{d_b}(\mathsf{dgm}(\mathrm{Filt}(\mathbb{X}_\mu)), \widehat{\mathsf{dgm}}_n)\right] \leqslant C' n^{-2/k} \tag{4.3}$$

where the infimum is taken over all the estimators of the persistence diagram.

5. Experiments

A series of experiments were conducted in order to illustrate the behavior of the persistence diagrams under sampling of metric spaces endowed with a probability measure and to compare the convergence performance obtained in practice with the theoretical results obtained in the previous sections.

5.1 Spaces and Data

We consider four different metric spaces, denoted $\mathbb{M}_1$, $\mathbb{M}_2$, $\mathbb{M}_3$ and $\mathbb{M}_4$ hereafter, that are described below.

- $\mathbb{M}_1$ **(Lissajous curve in $\mathbb{R}^2$):** the planar curve with the parametric equations $x(t) = \sin(3t + \pi/2)$, $y(t) = \sin(2t)$, $t \in [0, 2\pi]$ (see Figure 6, left). Its metric is the restriction of the Euclidean metric in $\mathbb{R}^2$ and it is endowed with the push forward by the parametrization of the uniform measure on the interval $[0, 2\pi]$.

- $\mathbb{M}_2$ **(sphere in $\mathbb{R}^3$):** the unit sphere in $\mathbb{R}^3$ (see Figure 6, center). Its metric is the restriction of the Euclidean metric in $\mathbb{R}^3$ and it is endowed with the uniform area measure on the sphere.

- $\mathbb{M}_3$ **(torus in $\mathbb{R}^3$):** the torus of revolution in $\mathbb{R}^3$ with the parametric equations $x(u, v) = (5 + \cos(u))\cos(v)$, $y(u, v) = (5 + \cos(u))\sin(v)$ and $z(u, v) = \sin(u)$, $(u, v) \in [0, 2\pi]^2$ (see Figure 6, right). Its metric is the restriction of the Euclidean metric in $\mathbb{R}^3$ and it is endowed with the push forward by the parametrization of the uniform measure on the square $[0, 2\pi]^2$.

- $\mathbb{M}_4$ **(rotating shape space):** for this space we used a 3D character from the SCAPE database (Anguelov et al., 2005) and considered all the images of this character from a view rotating around it. We converted these images in gray color and resized these images to $300 \times 400 = 120,000$ pixels (see Figure 7). Each is then identified with a point in $\mathbb{R}^{120,000}$ where the i^{th} coordinate is the level of gray of the i^{th} pixel. Moreover, we normalized these images by projecting them on the unit sphere in $\mathbb{R}^{120,000}$. The metric space $\mathbb{M}_4$ is the obtained subset of the unit sphere with the restriction of the Euclidean metric in $\mathbb{R}^{120,000}$. As it is parametrized by a circular set of views, it is endowed with the push forward of the uniform measure on the circle.

5.2 The Experiments

From each of the measured metric spaces $\mathbb{M}_1$, $\mathbb{M}_2$, $\mathbb{M}_3$ and $\mathbb{M}_4$ we sampled k sets of n points for different values of n from which we computed persistence diagrams for different geometric complexes (see Table 1). For $\mathbb{M}_1$, $\mathbb{M}_2$ and $\mathbb{M}_3$ we have computed the persistence diagrams for the 1 or 2-dimensional homology of the α-complex built on top of the sampled sets. As α-complexes have the same homotopy type as the corresponding union of balls, these persistence diagrams are the ones of the distance function to the sampled point set (Edelsbrunner, 1995). So, for each n we computed the average bottleneck distance between the obtained diagrams and the persistence diagram of the distance to the metric space from

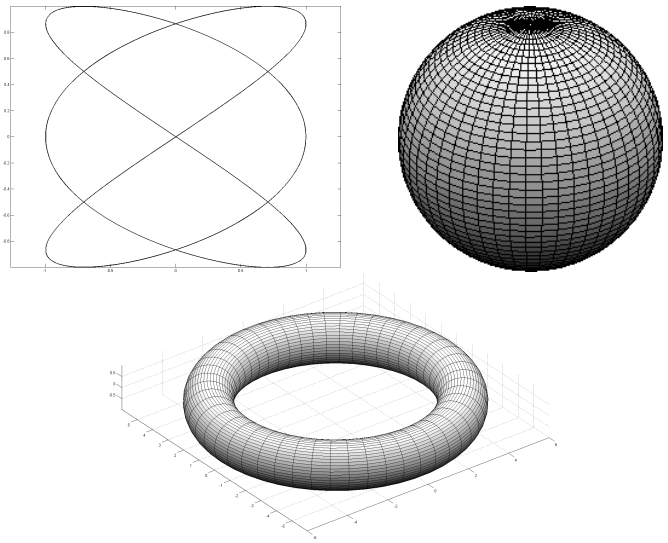


Figure 6: The spaces $\mathbb{M}_1$, $\mathbb{M}_2$ and $\mathbb{M}_3$.

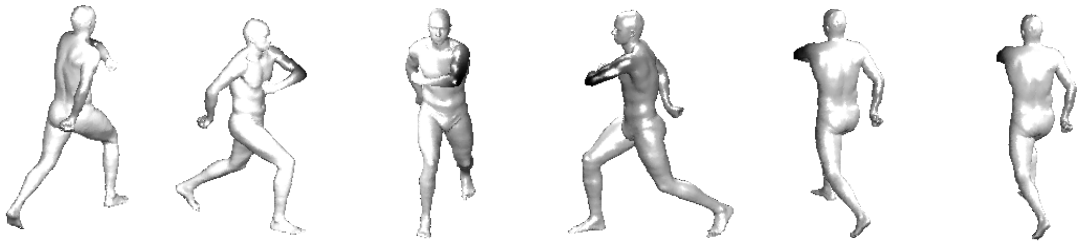


Figure 7: Images sampled from the space $\mathbb{M}_4$.

which the points were sampled. For $\mathbb{M}_4$, as it is embedded in a very high dimensional space, computing the α-complex is practically out of reach. So we have computed the persistence diagrams for the 1-dimensional homology of the Vietoris-Rips complex built on top of the sampled sets. The obtained results are described and discussed below.

- **Results for $\mathbb{M}_1$:** we approximated the 1-dimensional homology persistence diagram of the distance function to the Lissajous curve $\mathsf{dgm}(\mathbb{M}_1)$ by sampling $\mathbb{M}_1$ with $500,000$ points and computing the persistence diagram of the corresponding α-complex. As the Hausdorff distance between our sample and $\mathbb{M}_1$ was of order 10^{-5} we obtained a sufficiently precise approximation of $\mathsf{dgm}(\mathbb{M}_1)$ for our purpose. $\mathsf{dgm}(\mathbb{M}_1)$ is represented in blue on the left of Figure 8. For each n, the average bottleneck distance between $\mathsf{dgm}(\mathbb{M}_1)$ and the persistence diagrams obtained for the $k = 300$ randomly sampled sets $\mathbb{X}_n$ of size n has been used as an estimate $\hat{\mathbb{E}}$ of $\mathbb{E}\left[\mathrm{d_b}(\mathsf{dgm}(\mathrm{C}_\alpha(\mathbb{M}_1)), \mathsf{dgm}(\mathrm{C}_\alpha(\widehat{\mathbb{X}}_n)))\right]$ where C_α denotes the α-complex filtration. On Figure 8, right, $\log(\hat{\mathbb{E}})$ is plotted as a function of $\log(\log(n)/n)$. As expected, since the Lissajous curve is 1-dimensional, the points are close to a line of slope 1.

- **Results for $\mathbb{M}_2$ and $\mathbb{M}_3$:** the persistence diagrams $\mathsf{dgm}(\mathbb{M}_2)$ and $\mathsf{dgm}(\mathbb{M}_2)$ of the distance functions to $\mathbb{M}_2$ and $\mathbb{M}_3$ are known exactly and are represented in blue on Figures 9 and 10, left, respectively. Notice that we considered the 2-dimensional homology for $\mathbb{M}_2$ and 1-dimensional homology for $\mathbb{M}_3$. For $i = 2, 3$ and for each n, the average bottleneck distance between $\mathsf{dgm}(\mathbb{M}_i)$ and the persistence diagrams obtained for the $k = 100$ randomly sampled sets $\mathbb{X}_n$ of size n has been used as an estimate $\hat{\mathbb{E}}$ of $\mathbb{E}\left[\mathrm{d_b}(\mathsf{dgm}(\mathrm{C}_\alpha(\mathbb{M}_i)), \mathsf{dgm}(\mathrm{C}_\alpha(\widehat{\mathbb{X}}_n)))\right]$ where C_α denotes the α-complex filtration. $\log(\hat{\mathbb{E}})$ is plotted as a function of $\log(\log(n)/n)$ on Figures 9 and 10, right. As expected, since the sphere and the torus are 2-dimensional, the points are close to a line of slope 1/2.

- **Results for $\mathbb{M}_4$:** As in that case we do not know the persistence diagram of the Vietoris-Rips filtration built on top of $\mathbb{M}_4$, we only computed the 1-dimensional homology persistence diagrams of the Vietoris-Rips filtrations built on top of 20 sets of 250 points each, randomly sampled on $\mathbb{M}_4$. All these diagrams have been plotted on the same Figure 11, left. The right of Figure 11 represents a 2D embedding of one of the 250 points sampled data set using the Multidimensional Scaling algorithm (MDS). Since $\mathbb{M}_4$ is a set of images taken according a rotating point of view, it carries a cycle structure. This structure is reflected in the persistence diagrams that all have one point which is clearly off the diagonal. Notice also a second point off the diagonal which is much closer to it and that probably corresponds to the pinching in $\mathbb{M}_4$ visible at the bottom left of the MDS projection.

6. Discussion and Future Works

In previous works, the use of persistent homology in TDA has been mainly considered with a deterministic approach. As a consequence persistence diagrams were usually used as exploratory tools to analyze the topological structure of data. In this paper, we propose

Space	k **(sampled sets for each** n**)**	n **range**	**Geometric complex**
$\mathbb{M}_1$	300	$[2100 : 100 : 3000]$	α-complex
$\mathbb{M}_2$	100	$[12000 : 1000 : 21000]$	α-complex
$\mathbb{M}_3$	100	$[4000 : 500 : 8500]$	α-complex
$\mathbb{M}_4$	20	250	Vietoris-Rips complex

Table 1: Sampling parameters and geometric complexes where $[n_1 : h : n_2]$ denotes the set of integers $\{n_1, n_1 + h, n_1 + 2h, \cdots n_2\}$.

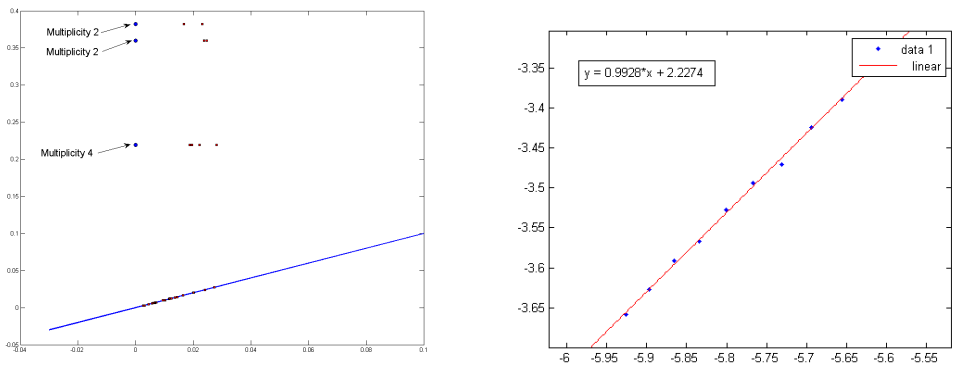


Figure 8: Convergence rate for the persistence diagram of the α-filtration built on top of points sampled on $\mathbb{M}_1$. Left: in blue the persistence diagram $\mathsf{dgm}(\mathbb{M}_1)$ of the distance to $\mathbb{M}_1$ (1-dimensional homology); in red a persistence diagram of the α-filtration built on top of $n = 2100$ points randomly sampled on $\mathbb{M}_1$. Right: the x-axis is $\log(\log(n)/n)$ where n is the number of points sampled on $\mathbb{M}_1$. The y-axis is the log of the estimated expectation of the bottleneck distance between the diagram obtained from an α-filtration built on top of n points sampled on $\mathbb{M}_1$ and $\mathsf{dgm}(\mathbb{M}_1)$.

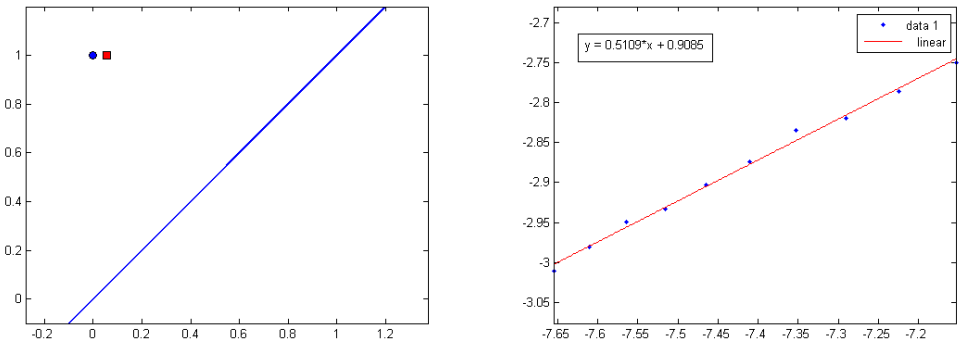


Figure 9: Convergence rate for the persistence diagram of the α-filtration built on top of points sampled on $\mathbb{M}_2$. Left: in blue the persistence diagram $\mathsf{dgm}(\mathbb{M}_2)$ of the distance to $\mathbb{M}_2$ (2-dimensional homology); in red a persistence diagram of the α-filtration built on top of $n = 12000$ points randomly sampled on $\mathbb{M}_2$. Right: the x-axis is $\log(\log(n)/n)$ where n is the number of points sampled on $\mathbb{M}_2$. The y-axis is the log of the estimated expectation of the bottleneck distance between the diagram obtained from an α-filtration built on top of n points sampled on $\mathbb{M}_2$ and $\mathsf{dgm}(\mathbb{M}_2)$.

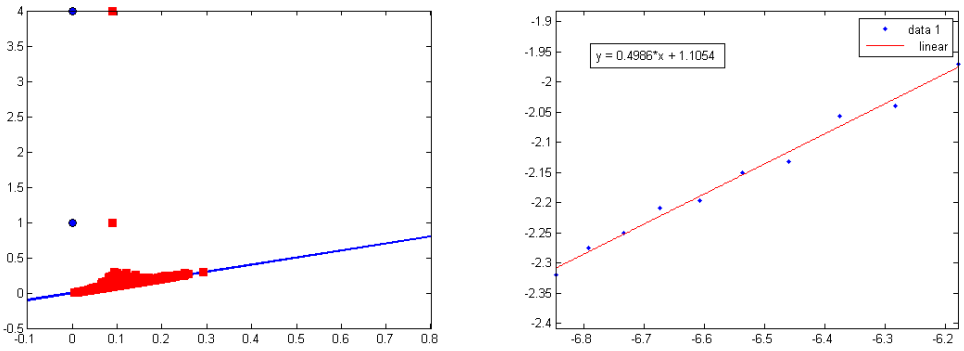


Figure 10: Convergence rate for the persistence diagram of the α-filtration built on top of points sampled on $\mathbb{M}_3$. Left: in blue the persistence diagram $\mathsf{dgm}(\mathbb{M}_3)$ of the distance to $\mathbb{M}_3$ (1-dimensional homology); in red a persistence diagram of the α-filtration built on top of $n = 14000$ points randomly sampled on $\mathbb{M}_3$. Right: the x-axis is $\log(\log(n)/n)$ where n is the number of points sampled on $\mathbb{M}_3$. The y-axis is the log of the estimated expectation of the bottleneck distance between the diagram obtain from α-filtration built on top of n points sampled on $\mathbb{M}_3$ and $\mathsf{dgm}(\mathbb{M}_3)$.

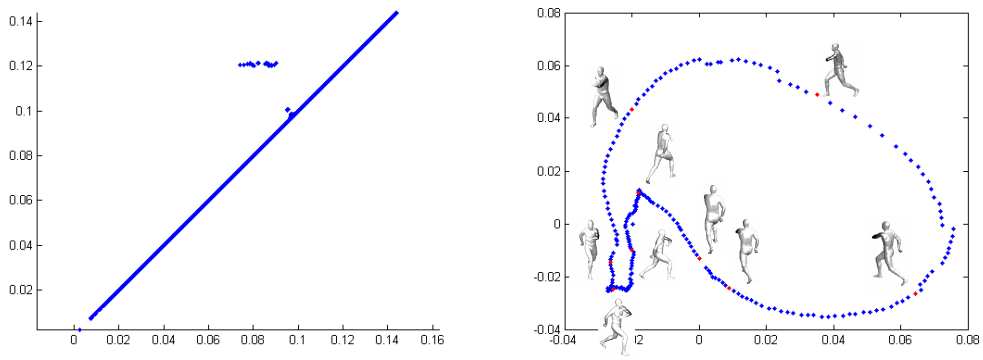


Figure 11: Left: on the same figure the 1-dimensional homology persistence diagrams of the Vietoris-Rips filtration of 20 sets of 250 points sampled on $\mathbb{M}_4$. Right: the plot of the embedding of $\mathbb{M}_4$ in $\mathbb{R}^2$ using MDS.

a rigorous framework to study the statistical properties of persistent homology and more precisely we give a general approach to study the rates of convergence for the estimation of persistence diagrams. The results we obtain open the door to a rigorous use of persistence diagrams in statistical framework. Our approach, consisting in reducing persistence diagram estimation to another more classical estimation problem (here support estimation) is based upon recently proven stability results in persistence theory that are very general.

In this paper, the persistence diagram of interest is the one of the support of the measure μ according which the data points are sampled. As a consequence, if the data points are sampled according to some perturbated measure ν whose support is not close to the one of μ then the estimator obviously non longer converges to the diagram of the support of μ. A first solution to overcome this problem is to plug denoising methods for support estimation (with respect to Hausdorff distance), such as deconvolution methods (Meister, 2009), to our approach.

Building on ideas developed by Chazal et al. (2011) and Caillerie et al. (2011), more satisfactory solutions have been recently proposed by Chazal et al. (2014a,b) that allow to infer persistent homology information from data corrupted by different kind of noise.

In another direction, an interesting representation of persistence diagrams as elements of a Hilbert space has recently been proposed by Bubenik (2012). Our results easily extend to this representation of persistence diagrams called *persistence landscapes*. Following this promising point of view, we also intend to adapt classical kernel-based methods with kernels carrying topological information.

Acknowledgments

The authors acknowledge the support of the European project CG-Learning EC contract No. 255827, the ANR projects GIGA (ANR-09-BLAN-0331-01) and TopData (ANR-13-BS01-0008), the ERC project GUDHI and the Google Faculty Research Award.

Appendix A. Lecam's Lemma

The version of Lecam's Lemma given below is from Yu (1997) (see also Genovese et al., 2012a). Recall that the total variation distance between two distributions P_0 and P_1 on a measured space $(\mathcal{X}, \mathcal{B})$ is defined by

$$\mathrm{TV}(P_0, P_1) = \sup_{B \in \mathcal{B}} |P_0(B) - P_1(B)|.$$

Moreover, if P_0 and P_1 have densities p_0 and p_1 for the same measure λ on $\mathcal{X}$, then

$$\mathrm{TV}(P_0, P_1) = \frac{1}{2}\ell_1(p_0, p_1) := \int_{\mathcal{X}} |p_0 - p_1| d\lambda.$$

Lemma 9 *Let $\mathcal{P}$ be a set of distributions. For $P \in \mathcal{P}$, let $\theta(P)$ take values in a metric space $(\mathbb{X}, \rho)$. Let P_0 and P_1 in $\mathcal{P}$ be any pair of distributions. Let $X_1, \ldots, X_n$ be drawn i.i.d. from some $P \in \mathcal{P}$. Let $\hat{\theta} = \hat{\theta}(X_1, \ldots, X_n)$ be any estimator of $\theta(P)$, then*

$$\sup_{P \in \mathcal{P}} \mathbb{E}_{P^n} \rho(\theta, \hat{\theta}) \geqslant \frac{1}{8} \rho\left(\theta(P_0), \theta(P_1)\right) \left[1 - \mathrm{TV}(P_0, P_1)\right]^{2n}.$$

Appendix B. Proofs

All the proofs of the paper are given in this section.

B.1 Proof of Theorem 2

The proof follows the lines of the proof of Cuevas and Rodríguez-Casal (2004, Theorem 3). The only point to be checked is that the covering number of $\mathbb{X}_\mu$ under the (a, b)-standard assumption can be controlled as when $b = d \in \mathbb{N}$, the rest of the proof being unchanged.

The covering number $\mathrm{cv}(\mathbb{X}_\mu, r)$ of $\mathbb{X}_\mu$ is the minimum number of balls of radius r that are necessary to cover $\mathbb{X}_\mu$:

$$\mathrm{cv}(\mathbb{X}_\mu, r) = \min \left\{ k \in \mathbb{N}^* \,:\, \exists (x_1, \ldots, x_k) \in (\mathbb{X}_\mu)^k \text{ such that } \mathbb{X}_\mu = \bigcup_{i=1}^{k} B(X_i, r) \right\}.$$

The packing number $\mathrm{pk}(\mathbb{X}_\mu, r)$ is the maximum number of balls of radius r that can be packed in $\mathbb{X}_\mu$ without overlap:

$$\mathrm{pk}(\mathbb{X}_\mu, r) = \max \left\{ \begin{array}{r} k \in \mathbb{N}^* \,:\, \exists (x_1, \ldots, x_k) \in (\mathbb{X}_\mu)^k \text{ such that } B(x_i, r) \subset \mathbb{X}_\mu \\ \text{and, } \forall i \neq j,\ B(x_i, r) \cap B(x_j, r) = \varnothing \end{array} \right\}$$

The covering and packing numbers are related by the following inequalities (see for instance Massart, 2007, p. 71):

$$\mathrm{pk}(\mathbb{X}_\mu, 2r) \leqslant \mathrm{cv}(\mathbb{X}_\mu, 2r) \leqslant \mathrm{pk}(\mathbb{X}_\mu, r). \tag{B.1}$$

Lemma 10 *Assume that the probability μ satisfies a standard (a, b)-assumption. Then for any $r > 0$ we have*

$$\mathrm{pk}(\mathbb{X}_\mu, r) \leqslant \frac{1}{a r^b} \vee 1 \text{ and } \mathrm{cv}(\mathbb{X}_\mu, r) \leqslant \frac{2^b}{a r^b} \vee 1.$$

Proof The result is trivial for $r \geqslant a^{-1/b}$. Let $r < a^{-1/b}$ and let $p = \mathrm{pk}(\mathbb{X}_\mu, r)$, we choose a maximal packing $B_1 = B(x_1, r), \cdots, B_p = B(x_p, r)$ of $\mathbb{X}_\mu$. Since the balls of the packing are pairwise disjoint and μ is a probability measure we have $\sum_{i=1}^p \mu(B_i) \leqslant 1$. Using that $\mu(B_i) \geqslant a r^b$ we obtain that $par^b \leqslant \sum_{i=1}^p \mu(B_i) \leqslant 1$ from which we get the upper bound on $\mathrm{pk}(\mathbb{X}_\mu, r)$. Since from (B.1) we have $\mathrm{cv}(\mathbb{X}_\mu, r) \leqslant \mathrm{pk}(\mathbb{X}_\mu, r/2)$ we immediately deduce the upper bound on $\mathrm{cv}(\mathbb{X}_\mu, r)$. ■

B.2 Proof of Proposition 4

We first prove the upper bound.

B.2.1 Upper Bound

According to Corollary 3, thanks to Fubini we have

$$\mathbb{E}\left[\mathrm{d_b}(\mathsf{dgm}(\mathrm{Filt}(\mathbb{X}_\mu)), \mathsf{dgm}(\mathrm{Filt}(\widehat{\mathbb{X}}_n)))\right] \leqslant \int_{\varepsilon>0} \mathbb{P}\left[\mathrm{d_b}(\mathsf{dgm}(\mathrm{Filt}(\mathbb{X}_\mu)), \mathsf{dgm}(\mathrm{Filt}(\widehat{\mathbb{X}}_n))) > \varepsilon\right] d\varepsilon$$

Let $\varepsilon_n = 4\left(\frac{\log n}{an}\right)^{1/b}$. By bounding the probability inside this integral by one on $[0, \varepsilon_n]$, we find that:

$$\begin{aligned}\mathbb{E}\left[\mathrm{d_b}(\mathsf{dgm}(\mathrm{Filt}(\mathbb{X}_\mu)), \mathsf{dgm}(\mathrm{Filt}(\widehat{\mathbb{X}}_n)))\right] &\leqslant \varepsilon_n + \int_{\varepsilon>\varepsilon_n} \frac{8^b}{a}\varepsilon^{-b}\exp(-na\varepsilon^b/4^b)d\varepsilon \\ &\leqslant \varepsilon_n + \frac{4n2^b}{b}(na)^{-1/b}\int_{u\geqslant \log n} u^{1/b-2}\exp(-u)du.\end{aligned}$$

Now, if $b \geqslant \frac{1}{2}$ then $u^{1/b-2} \leqslant (\log n)^{1/b-2}$ for any $u \geqslant \log n$ and then

$$\begin{aligned}\mathbb{E}\left[\mathrm{d_b}(\mathsf{dgm}(\mathrm{Filt}(\mathbb{X}_\mu)), \mathsf{dgm}(\mathrm{Filt}(\widehat{\mathbb{X}}_n)))\right] &\leqslant \varepsilon_n + 4\frac{2^b}{b}\left(\frac{\log n}{n}\right)^{1/b}(\log n)^{-2} \\ &\leqslant C_1(a,b)\left(\frac{\log n}{n}\right)^{1/b} \qquad \text{(B.2)}\end{aligned}$$

where the constant $C_1(a,b)$ only depends on a and b. If $0 < b < \frac{1}{2}$, let $p := \lfloor \frac{1}{b} \rfloor$ and then

$$\begin{aligned}\int_{u\geqslant u_n:=\log n} u^{1/b-2}\exp(-u)du &= u_n^{1/b-2}\exp(u_n) + (\frac{1}{b}-2)u_n^{1/b-3}\exp(u_n) + \cdots + \\ &\quad + \prod_{i=2}^{p}\left(\frac{1}{b}-i\right)u_n^{1/b-p}\exp(u_n) + \int_{u\geqslant \log n} u^{1/b-p-1}\exp(-u)du \\ &\leqslant C_2(a,b)\frac{(\log n)^{1/b-2}}{n}\end{aligned}$$

where $C_2(a,b)$ only depends on a and b. Thus (B.2) is also satisfied for $b < \frac{1}{2}$ and the upper bound is proved.

B.2.2 Lower Bound

To prove the lower bound, it will be sufficient to consider two Dirac distributions. We take for $P_{0,n} = P_x$ the Dirac distribution on $\mathbb{X}_0 := \{x\}$ and it is clear that $P_0 \in \mathcal{P}(a, b, \mathbb{M})$. Let $P_{1,n}$ be the distribution $\frac{1}{n}\delta_{x_n} + (1-\frac{1}{n})P_0$. The support of $P_{1,n}$ is denoted $\mathbb{X}_{1,n} := \{x\} \cup \{x_n\}$. Note that for any $n \geqslant 2$ and any $r \leqslant \rho(x, x_n)$:

$$P_{1,n}\left(B(x,r)\right) = 1 - \frac{1}{n} \geqslant \frac{1}{2} \geqslant \frac{1}{2\rho(x,x_n)^b} r^b \geqslant a r^b$$

and

$$P_{1,n}\left(B(x_n,r)\right) = \frac{1}{n} = \frac{1}{n\rho(x,x_n)^b} r^b \geqslant a r^b.$$

Moreover, for $r > \rho(x, x_n)$, $P_{1,n}\left(B(0,r)\right) = P_{1,n}\left(B(x_n,r)\right) = 1$. Thus for any $r > 0$ and any $x \in \mathbb{X}_{1,n}$:

$$P_{1,n}\left(B(x,r)\right) \geqslant a r^b \wedge 1$$

and $P_{1,n}$ also belongs to $\mathcal{P}(a, b, \mathbb{M})$.

The probability measure P_0 is absolutely continuous with respect to $P_{1,n}$ and the density of P_0 with respect to $P_{1,n}$ is $p_{0,n} := \frac{n}{n-1}\mathbb{1}_{\{x\}}$. Then

$$\begin{aligned} TV(P_0, P_{1,n}) &= \int_{\mathbb{M}} |1 - \frac{n}{n-1}\mathbb{1}_{\{x\}}| \, dP_{1,n} \\ &= \frac{2}{n}. \end{aligned}$$

Next, $\left[1 - TV(P_0, P_{1,n})\right]^{2n} = (1 - \frac{2}{n})^{2n} \to e^{-4}$ as n tends to infinity. It remains to compute $\mathrm{d_b}(\mathsf{dgm}(\mathrm{Filt}(\mathbb{X}_0)), \mathsf{dgm}(\mathrm{Filt}(\mathbb{X}_{1,n})))$. We only consider here the Rips case, the other filtrations can be treated in a similar way. The barcode of $\mathrm{Filt}(\mathbb{X}_0)$ is composed of only one segment $(0, +\infty)$ for the 0-cycles. The barcode of $\mathrm{Filt}(\mathbb{X}_{1,n})$ is composed of the segment of $\mathrm{Filt}(\mathbb{X}_0)$ and one more 0-cycle : $(0, \rho(x, x_n))$. Thus we have:

$$\begin{aligned} \mathrm{d_b}(\mathsf{dgm}(\mathrm{Filt}(\mathbb{X}_0)), \mathsf{dgm}(\mathrm{Filt}(\mathbb{X}_{1,n}))) &= d_\infty\left(\Delta, (0, \rho(x, x_n))\right) \\ &= \frac{\rho(x, x_n)}{2}. \end{aligned}$$

The proof is then complete using Lecam's Lemma (Lemma 9).

B.3 Proof of Theorem 5

Let A be the interval $[0, 1]$ and c a positive constant to be chosen further. We consider k "holes" H_i of length $c\frac{\log n}{n}$ each, distant enough from each other that we remain $(\frac{1}{2}, 1)$-standard when we remove any number of H_i from A, which is possible as long as $k\, c\frac{\log n}{n} < \frac{1}{2}$. We denote $A_i = A \backslash H_i$. For $I \subset \{1, \ldots, k\}$, $A_I = \bigcap_{i \in I} A_i$, $B = A^n$, $B_I = A_I^n$. Denoting the uniform measure on $[0, 1]$ by λ, we have $\lambda^{\otimes n}(B_I) = (1 - |I| c\frac{\log n}{n})^n \sim n^{-|I|c}$.

The main idea is that when sampling n points from A, most likely (at least) one of the H_i contains no points. Without points in H_i, the estimator cannot distinguish A from A_i, but since those two have diagrams at distance $c\frac{\log n}{n}$, this gives a bound on the quality of the estimator. The technical difficulty is that several H_i can be empty at the same time.

For a given n, let $\widehat{\mathsf{dgm}}_n$ be an estimator of persistence diagram of the sampling distribution support. Assume for the moment that $\widehat{\mathsf{dgm}}_n$ satisfies

$$\sup_{\mu\in\mathcal{P}(\frac{1}{2},1,[0,1])} \mathbb{E}\left[\mathrm{d_b}\left(\mathsf{dgm}(\mathrm{Filt}(\mathbb{X}_\mu)),\widehat{\mathsf{dgm}}_n\right)\right] \leqslant \frac{1}{2}\frac{\log n}{n}. \tag{B.3}$$

Under this assumption, our goal is to lower bound $\mathbb{E}\left[\mathrm{d_b}\left(\mathsf{dgm}(\mathrm{Filt}(\mathbb{X}_\mu)),\widehat{\mathsf{dgm}}_n\right)\right]$ for μ equal to λ the uniform distribution on $[0,1]$. The estimator $\widehat{\mathsf{dgm}}_n$ can also be written as $\widehat{\mathsf{dgm}}_n = g(X_1,\dots,X_n)$ where g is a measurable application from B into the set of persistence diagrams. First we note that when the observations are sampled according to λ:

$$\begin{aligned}\mathbb{E}\left[\mathrm{d_b}\left(\mathsf{dgm}(\mathrm{Filt}(\mathbb{X}_\lambda)),\widehat{\mathsf{dgm}}_n\right)\right] &= \int_B \mathrm{d_b}\left(\mathsf{dgm}(A),g(x)\right)\, d\lambda^{\otimes n}(x)\\ &\geqslant \int_{\bigcup_{1\leqslant i\leqslant k} B_i} \mathrm{d_b}\left(\mathsf{dgm}(A),g(x)\right)\, d\lambda^{\otimes n}(x) =: R_n\end{aligned}$$

so it will be sufficient to bound this last integral. Applying Inequality B.3 to the uniform distribution μ_I on the set A_I, we find that

$$\frac{1}{\lambda^{\otimes n}(B_I)}\int_{B_I} d_B\left(\mathsf{dgm}(A_I),g(x)\right)\, d\lambda^{\otimes n}(x) \leqslant \frac{1}{2}\frac{\log n}{n}.$$

Let $M_I := \int_{B_I} \mathrm{d_b}\left(\mathsf{dgm}(A),g(x)\right)\, d\lambda^{\otimes n}(x)$. Knowing that $\mathrm{d_b}(\mathsf{dgm}(A),\mathsf{dgm}(A_I)) = c\frac{\log n}{n}$ and using the triangular inequality, we find that

$$\left|\frac{M_I}{\lambda^{\otimes n}(B_I)} - c\frac{\log n}{n}\right| \leqslant \frac{1}{2}\frac{\log n}{n}. \tag{B.4}$$

By applying the inclusion-exclusion principle for the union of the B_i's, we find that $R_n \geqslant R_{1,n} - R_{2,,n}$ where $R_{1,n} = \sum_i M_i$ and $R_{2,n} = \sum_{i<j} M_{\{i,j\}}$. According to (B.4) we have

$$R_{1,n} \geqslant k\left(c-\frac{1}{2}\right)\frac{\log n}{n}\left(1-c\frac{\log n}{n}\right)^n$$

and

$$R_{2,n} \leqslant \frac{k(k-1)}{2}\left(c+\frac{1}{2}\right)\frac{\log n}{n}\left(1-2c\frac{\log n}{n}\right)^n.$$

We take $c=\frac{3}{4}$. Then the lower bound of $R_{1,n}$ is equivalent to $\frac{k}{4}\frac{\log n}{n}n^{-3/4}$ and

$$\frac{R_{2,n}}{R_{1,n}} \leqslant 5\frac{k-1}{2}\frac{\left(1-\frac{3}{2}\frac{\log n}{n}\right)^n}{\left(1-\frac{3}{4}\frac{\log n}{n}\right)^n} \sim_{n\to\infty} 5\frac{k-1}{2}n^{-\frac{3}{4}}.$$

We take $k = k_n := \left\lfloor \frac{n^{3/4}}{5}\right\rfloor$ in order to have $\frac{R_{2,n}}{R_{1,n}}$ tending to $\frac{1}{2}$ as n tends to infinity. We thus have

$$\liminf_n \frac{n}{\log n} R_n \geqslant \frac{1}{40}.$$

Moreover, note that

$$k_n c\frac{\log n}{n} \sim \frac{3}{20}n^{3/4}\frac{\log n}{n}$$

which is smaller than $\frac{1}{2}$ for n large enough so the $(\frac{1}{2},1)$-standard assumption is verified for n large enough.

To summarize, for any n and any estimator $\widehat{\mathsf{dgm}}_n$: either $\widehat{\mathsf{dgm}}_n$ satisfies (B.3) and then

$$\sup_{\mu\in\mathcal{P}(\frac{1}{2},1,[0,1])} \mathbb{E}\left[\mathrm{d_b}\left(\mathsf{dgm}(\mathrm{Filt}(\mathbb{X}_\mu)),\widehat{\mathsf{dgm}}_n\right)\right] \geqslant R_n,$$

or

$$\sup_{\mu\in\mathcal{P}(\frac{1}{2},1,[0,1])} \mathbb{E}\left[\mathrm{d_b}\left(\mathsf{dgm}(\mathrm{Filt}(\mathbb{X}_\mu)),\widehat{\mathsf{dgm}}_n\right)\right] \geqslant \frac{1}{2}\frac{\log n}{n}.$$

Finally we have that for any estimator $\widehat{\mathsf{dgm}}_n$:

$$\liminf_n \frac{n}{\log n}\sup_{\mu\in\mathcal{P}(\frac{1}{2},1,[0,1])} \mathbb{E}\left[\mathrm{d_b}\left(\mathsf{dgm}(\mathrm{Filt}(\mathbb{X}_\mu)),\widehat{\mathsf{dgm}}_n\right)\right] \geqslant \min\left(\frac{1}{2},\liminf_n \frac{n}{\log n}R_n\right) \geqslant \frac{1}{40}$$

and the theorem is proved.

B.4 Proofs for Section 4.1

Lemma 11 *1. Under assumption $[B]$, we have $\mathrm{d}_H(G_0,\mathbb{X}_f)=0$.*

2. Under Assumptions $[A]$ and $[B]$, μ satisfies a standard assumption with $b=\alpha+k$ and with a depending on $\mathcal{F}(\alpha)$.

Proof First, note that we always have

$$\overset{\circ}{G_0} \subset \mathbb{X}_f \subset \overline{G_0}. \tag{B.5}$$

Indeed, if $\overset{\circ}{G_0}\cap(\chi\backslash\mathbb{X}_f)$ is non empty, let x be in the intersection. Then there exists $\varepsilon>0$ such that $B(x,\varepsilon)\subset G_0$ and $B(x,\varepsilon)\subset(\chi\backslash\mathbb{X}_f)$ since $\mathbb{X}_f$ is assumed to be closed. The first inclusion then gives that $\mu(B(x,\varepsilon))>0$ whereas the second inclusion gives that $\mu(B(x,\varepsilon))=0$. Thus $\overset{\circ}{G_0}\cap(\chi\backslash\mathbb{X}_f)$ is empty, the second inclusion in (B.5) is obvious since $\mathbb{X}_f$ is assumed to be closed.

Then,

$$\begin{aligned}
\mathrm{d_H}(\mathbb{X}_f,G_0) &= \max(\sup_{x\in\mathbb{X}_f} d(x,G_0),\sup_{x\in G_0} d(x,\mathbb{X}_f))\\
&= \max(\sup_{x\in\mathbb{X}_f} d(x,\overline{G_0}),\sup_{x\in\overline{G_0}} d(x,\mathbb{X}_f))\\
&= \sup_{x\in\overline{G_0}} d(x,\mathbb{X}_f)\\
&= \sup_{x\in\partial G_0} d(x,\mathbb{X}_f)
\end{aligned} \tag{B.6}$$

where we use the continuity of the distance function for the second equality and (B.5) for the two last ones. It follows from assumption [B] that for any $x \in \partial G_0$, $d(x, \overset{\circ}{G_0}) = 0$. Thus $d(x, \mathbb{X}_f) = 0$ according to (B.5) and we have proved that (B.6) is equal to zero.

We now prove the second point of the Lemma. Let $x \in \bar{G}_0$ and let $r > 0$ such that

$$\frac{r}{2}\left(1 \wedge \frac{1}{C_b}\right) < \varepsilon_0 \wedge \left(\frac{\delta_a}{C_a}\right)^{1/\alpha}. \tag{B.7}$$

According to Assumption [B], for $\varepsilon = \frac{r}{2}\left(1 \wedge \frac{1}{C_b}\right)$, there exists $y \in I_\varepsilon(G_0)$ such that $d(x,y) \leqslant C_b\varepsilon \leqslant \frac{r}{2}$. Then, there exists $z \in I_\varepsilon$ such that $y \in B(z,\varepsilon) \subset I_\varepsilon$. Since $\varepsilon \leqslant \frac{r}{2}$ we find that $B(z,\varepsilon) \subset B(x,r) \cap G_0$. Thus,

$$\begin{aligned}
\mu\left(B(x,r)\right) &\geqslant \int_{B(z,\varepsilon)} f(u)\, d\lambda(u) \\
&\geqslant \int_{B(z,\varepsilon)} \delta_a \wedge C_a d(u, \partial G_0)^\alpha \, d\lambda(u) \\
&\geqslant C_a \int_{B(z,\varepsilon)} (\varepsilon - \|u - z\|)^\alpha \, d\lambda(u) \\
&\geqslant C_a s_{k-1} \int_0^\varepsilon (\varepsilon - r)^\alpha \, r^{k-1} \, dr
\end{aligned}$$

where s_{k-1} denotes the surface area of the unit $k-1$-sphere of $\mathbb{R}^k$, and where we have used Assumption [A] for the second inequality and the fact $C_a\varepsilon^\alpha \leqslant \delta_a$ for the third one. Finally we find that for any r satisfying (B.7):

$$\begin{aligned}
\mu\left(B(x,r)\right) &\geqslant \frac{C_a s_{k-1}(k-1)!}{(\alpha+1)\dots(\alpha+k)}\varepsilon^{\alpha+k} \\
&\geqslant \frac{C_a s_{k-1}(k-1)!(1 \wedge \frac{1}{C_b})^{\alpha+k}}{2^{\alpha+k}(\alpha+1)\dots(\alpha+k)} r^{\alpha+k}
\end{aligned}$$

and we obtain that μ satisfies that standard assumption with $b = \alpha + k$. ■

B.4.1 Proof of Proposition 6

The first point of the proposition is an immediate consequence of the first point of Theorem 4 together with Lemma 11. We now prove the lower bound by adapting some ideas from the proof of Proposition 3 in Singh et al. (2009) about the Hausdorff lower bound. At the price of loosing a logarithm term in the lower bound, we propose here a proof based on a two-alternative analysis.

The function f_0 is defined on χ as follows for $r_0 > 0$ small enough:

$$f_0 = \begin{cases} C_a\|x\|^\alpha & \text{if } \|x\| \leqslant r_0 \\ C_0 & \text{if } r_0 \leqslant \|x\| \leqslant 2r_0 \\ C_a(3r_0 - \|x\|)^\alpha & \text{if } 2r_0 \leqslant \|x\| \leqslant 3r_0 \\ 0 & \text{elsewhere} \end{cases}$$

where

$$C_0 = \frac{1 - C_a s_{k-1} r_0^{k+\alpha}(\frac{1}{k+\alpha} + I_\alpha)}{s_{k-1} r_0^k (2^k - 1)/k} \qquad \text{with } I_\alpha = \int_2^3 k^{k-1}(3-u)^\alpha du.$$

For $n \geqslant 1$ let $\varepsilon_n := n^{-1/(k+\alpha)}$, the function $f_{1,n}$ is defined on χ by

$$f_{1,n} = \begin{cases} \|x\|^\alpha & \text{if } \varepsilon_n \leqslant \|x\| \leqslant r_0 \\ C_{1,n} & \text{if } r_0 \leqslant \|x\| \leqslant 2r_0 \\ C_a(3r_0 - \|x\|)^\alpha & \text{if } 2r_0 \leqslant \|x\| \leqslant 3r_0 \\ 0 & \text{elsewhere} \end{cases}$$

where

$$\begin{aligned} C_{1,n} &= \frac{1 - C_a s_{k-1}\left\{r_0^{k+\alpha}(\frac{1}{k+\alpha} + I_\alpha) - \frac{\varepsilon_n^{k+\alpha}}{k+\alpha}\right\}}{s_{k-1} r_0^k (2^k - 1)/k} \\ &= C_0 + \frac{k C_a \varepsilon_n^{k+\alpha}}{(k+\alpha) r_0^k (2^k - 1)}. \end{aligned}$$

We assume that δ_a is small enough so that we can choose r_0 such that $\delta_a \leqslant C_0$ for n large enough. Then f_0 and $f_{1,n}$ are both densities and they both belong to $\mathcal{F}(\alpha)$ for n large enough. The support of $f_0 d\lambda$ is equal to $\mathbb{X}_0 := \bar{B}(0, 3r_0)$ whereas the support of $f_{1,n} d\lambda$ is equal to $\mathbb{X}_{1,n} = \bar{B}(0, 3r_0) \backslash \bar{B}(0, \varepsilon_n)$. Next,

$$\begin{aligned} \mathrm{TV}(f_0\, d\lambda, f_{1,n}\, d\lambda) &= \int_\chi |f_0 - f_{1,n}| dx \\ &= s_{k-1} C_a \int_0^{\varepsilon_n} r^{\alpha+k-1}\, dr + s_{k-1} \int_{r_0}^{2r_0} (C_{1,n} - C_0) r^{k-1} dr \\ &= \frac{2 s_{k-1} C_a}{k+\alpha} \varepsilon_n^{k+\alpha} \end{aligned}$$

Note that $(1 - \mathrm{TV}(f_0\, d\lambda, f_{1,n}\, d\lambda)]^{2n} \to \exp(-\frac{4 s_{k-1} C_a}{k+\alpha})$ as n tends to infinity. It remains to compute $\mathrm{d_b}(\mathsf{dgm}(\mathrm{Filt}(\mathbb{X}_0)), \mathsf{dgm}(\mathrm{Filt}(\mathbb{X}_{1,n})))$. We only consider here the Rips case, the other filtrations can be treated in a similar way. The barcode of $\mathrm{Filt}(\mathbb{X}_0)$ is composed of only one segment $(0, +\infty)$ for the 0-cycles. The barcode of $\mathrm{Filt}(\mathbb{X}_{1,n})$ is composed of the segment of $\mathrm{Filt}(\mathbb{X}_0)$ and one more 1-cycle : $(0, 2\varepsilon_n)$. Thus we have:

$$\begin{aligned} \mathrm{d_b}(\mathsf{dgm}(\mathrm{Filt}(\mathbb{X}_0)), \mathsf{dgm}(\mathrm{Filt}(\mathbb{X}_{1,n}))) &= d_\infty(\Delta, (0, \varepsilon)) \\ &= \varepsilon_n. \end{aligned}$$

We then finish the proof using Lecam's Lemma.

B.5 Proof of Proposition 8

We only need to prove the lower bound since the upper bound is a direct corollary of Theorem 3 in Genovese et al. (2012b). To prove the lower bound, we may use the particular manifolds defined in Genovese et al. (2012a) and also used by the same authors for the proof of Theorem 2 in Genovese et al. (2012b). Without loss of generality, we assume that

$\chi = [-L, L]^D$ and that $\kappa < L/2$. For $\ell \leqslant L$, let M and M' be the two manifolds of χ defined by

$$M = [-\ell, \ell]^D \cap \{x \in \chi \,|\, x_{k+1} = \cdots = x_D = 0\} \quad \text{and} \quad M' = 2\kappa e_{k+1} + M$$

where e_{k+1} is the $k+1$-th vector of the canonical basis in $\mathbb{R}^D$. We assume that ℓ is chosen so that $b < 2(2\ell)^{-k} < B$. Let μ_0 be the uniform measure on $\mathbb{X}_0 := M \cup M'$ and then $\mu_0 \in \mathcal{H}$.

According to Genovese et al. (2012a, Theorem 6), for $0 < \gamma < \kappa$, we can define a manifold M_γ which can be seen as a perturbation of M such that:

- $\Delta(M_\gamma) = \kappa$
- $\mathrm{d_H}(M_\gamma, M) = \gamma$ and $\mathrm{d_H}(M_\gamma, M') = 2\kappa - \gamma$
- If $A = \{x \in M_\gamma \,|\, x \notin M\}$ then $\mu_1(A) \leqslant C\gamma^{k/2}$ where $C > 0$ and where μ_1 is the uniform measure on $\mathbb{X}_1 := M_\gamma \cup M'$.

For small enough γ we see that μ_1 satisfies $[H_2]$ and thus $\mu_1 \in \mathcal{H}$.

As before, we only consider here filtrations of Rips complexes. The persistence diagrams of $\mathrm{Filt}(\mathbb{X}_0)$ and $\mathrm{Filt}(\mathbb{X}_1)$ are exactly the same except for the diagram of 0-cycles : the first filtration has a barcode with a segment $(0, 2\kappa)$ whereas the corresponding barcode for $\mathrm{Filt}(\mathbb{X}_1)$ is $(0, 2\kappa - \gamma)$. Thus, $\mathrm{d_b}(\mathrm{Filt}(\mathbb{X}_0), \mathrm{Filt}(\mathbb{X}_1)) = \gamma$. Moreover, $\mathrm{TV}(\mu_0, \mu_1) \leqslant |\mu_0(A) - \mu_1(A)| \leqslant C\gamma^{k/2}$. Finally, we choose $\gamma = (1/n)^{k/2}$ as in the proof of Genovese et al. (2012b, Theorem 2) and we conclude using Lecam's Lemma.

References

Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. SCAPE: shape completion and animation of people. In *Proceedings of SIGGRAPH*, pages 408–416, 2005.

Sivaraman Balakrishnan, Alessandro Rinaldo, Don Sheehy, Aarti Singh, and Larry A. Wasserman. Minimax rates for homology inference. *Journal of Machine Learning Research - Proceedings Track*, 22:64–72, 2012.

Ulrich Bauer, Axel Munk, Hannes Sieling, and Max Wardetzky. Persistent homology meets statistical inference - a case study: Detecting modes of one-dimensional signals. *ArXiv:1404.1214*, 2014.

Gérard Biau, Benoît Cadre, David M. Mason, and Bruno Pelletier. Asymptotic normality in density support estimation. *Electronic Journal of Probability*, 14:2617–2635, 2009.

Peter Bubenik. Statistical topology using persistence landscapes. *ArXiv:*, July 2012.

Peter Bubenik and Peter T Kim. A statistical approach to persistent homology. *Homology, Homotopy and Applications*, 9(2):337–362, 2007.

Dmitri Burago, Yuri Burago, and Sergei Ivanov. *A Course in Metric Geometry*, volume 33. American Mathematical Society Providence, 2001.

Claire Caillerie, Frédéric Chazal, Jérôme Dedecker, and Bertrand Michel. Deconvolution for the Wasserstein metric and geometric inference. *Electronic Journal of Statistics*, 5: 1394–1423, 2011.

Gunnar Carlsson. Topology and data. *AMS Bulletin*, 46(2):255–308, 2009.

Frédéric Chazal, David Cohen-Steiner, Marc Glisse, Leonidas J. Guibas, and Steve Y. Oudot. Proximity of persistence modules and their diagrams. In *SCG*, pages 237–246, 2009a.

Frédéric Chazal, David Cohen-Steiner, Leonidas J. Guibas, Facundo Mémoli, and Steve Y. Oudot. Gromov-hausdorff stable signatures for shapes using persistence. *Computer Graphics Forum (proc. SGP 2009)*, pages 1393–1403, 2009b.

Frédéric Chazal, David Cohen-Steiner, and Quentin Mérigot. Geometric inference for probability measures. *Foundations of Computational Mathematics*, 11(6):733–751, 2011.

Frédéric Chazal, Vin de Silva, Marc Glisse, and Steve Oudot. The structure and stability of persistence modules. *ArXiv:1207.3674*, 2012a.

Frédéric Chazal, Vin de Silva, and Steve Oudot. Persistence stability for geometric complexes. *ArXiv*, july 2012b.

Frédéric Chazal, Leonidas J. Guibas, Steve Y. Oudot, and Primoz Skraba. Persistence-based clustering in Riemannian manifolds. *J. ACM*, 60(6):41:1–41:38, November 2013.

Frédéric Chazal, Brittany Terese Fasy, Fabrizio Lecci, Bertrand Michel, Alessandro Rinaldo, and Larry Wasserman. Robust topological inference: Distance to a measure and kernel distance. *ArXiv:1412.7197*, 2014a.

Frédéric Chazal, Brittany Terese Fasy, Fabrizio Lecci, Bertrand Michel, Alessandro Rinaldo, and Larry Wasserman. Subsampling methods for persistent homology. *ArXiv:1406.1901*, 2014b.

David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of persistence diagrams. *Discrete & Computational Geometry*, 37(1):103–120, 2007.

Antonio Cuevas. Set estimation: another bridge between statistics and geometry. *Boletín de Estadística e Investigación Operativa*, 25(2):71–85, 2009.

Antonio Cuevas and Ricardo Fraiman. A plug-in approach to support estimation. *The Annals of Statistics*, 25(6):2300–2312, 1997.

Antonio Cuevas and Alberto Rodríguez-Casal. On boundary estimation. *Advances in Applied Probability*, 36(2):340–354, 2004.

Antonio Cuevas, Ricardo Fraiman, and Beatriz Pateiro-López. On statistical properties of sets fulfilling rolling-type conditions. *Advances in Applied Probability*, 44(2):311–329, 2012.

Vin De Silva and Robert Ghrist. Homological sensor networks. *Notices of the American Mathematical Society*, 54(1), 2007.

Ernesto De Vito, Lorenzo Rosasco, and Alessandro Toigo. Learning sets with separating kernels. *Applied and Computational Harmonic Analysis*, 37(2):185 – 217, 2014.

Luc Devroye and Gary L. Wise. Detection of abnormal behavior via nonparametric estimation of the support. *SIAM Journal on Applied Mathematics*, 38(3):480–488, 1980.

Lutz Dümbgen and Günther Walther. Rates of convergence for random approximations of convex sets. *Advances in Applied Probability*, 28(2):384–393, 1996.

Herbert Edelsbrunner. The union of balls and its dual shape. *Discrete & Computational Geometry*, 13(1):415–440, 1995.

Herbert Edelsbrunner and John L Harer. *Computational Topology: an Introduction*. American Mathematical Soc., 2010.

Herbert Edelsbrunner, David Letscher, and Afra Zomorodian. Topological persistence and simplification. *Discrete & Computational Geometry*, 28:511–533, 2002.

Brittany Terese Fasy, Fabrizio Lecci, Alessandro Rinaldo, Larry Wasserman, Sivaraman Balakrishnan, Aarti Singh, et al. Confidence sets for persistence diagrams. *The Annals of Statistics*, 42(6):2301–2339, 2014.

Herbert Federer. Curvature measures. *Transactions of the American Mathematical Society*, 93:418–491, 1959.

Christopher R. Genovese, Marco Perone-Pacifico, Isabella Verdinelli, and Larry Wasserman. Minimax manifold estimation. *Journal of Machine Learning Research*, 13:1263–1291, july 2012a.

Christopher R. Genovese, Marco Perone-Pacifico, Isabella Verdinelli, and Larry Wasserman. Manifold estimation and singular deconvolution under hausdorff loss. *The Annals of Statistics*, 40:941–963, 2012b.

Allen Hatcher. *Algebraic Topology*. Cambridge Univ. Press, 2001.

Peter M Kasson, Afra Zomorodian, Sanghyun Park, Nina Singhal, Leonidas J Guibas, and Vijay S Pande. Persistent voids: a new structural metric for membrane fusion. *Bioinformatics*, 23(14):1753–1759, 2007.

Aleksandr Petrovič Korostelëv and Alexandre B. Tsybakov. *Minimax Theory of Image Reconstruction*, volume 82 of *Lecture Notes in Statistics*. Springer-Verlag, New York, 1993.

Aleksandr Petrovič Korostelëv, Leopold Simar, and Alexandre B. Tsybakov. Efficient estimation of monotone boundaries. *Ann. Statist.*, 23(2):476–489, 1995.

Pascal Massart. *Concentration Inequalities and Model Selection*. Springer, Berlin, 2007. Lectures from the 33rd Summer School on Probability Theory held in Saint-Flour, July 6–23, 2003.

Alexander Meister. *Deconvolution Problems in Nonparametric Statistics*, volume 193. Springer, 2009.

Yuriy Mileyko, Sayan Mukherjee, and John Harer. Probability measures on the space of persistence diagrams. *Inverse Problems*, 27(12), 2011.

Partha Niyogi, Stephen Smale, and Shmuel Weinberger. Finding the homology of submanifolds with high confidence from random samples. *Discrete & Computational Geometry*, 39(1-3):419–441, March 2008.

Partha Niyogi, Stephen Smale, and Shmuel Weinberger. A topological view of unsupervised learning from noisy data. *SIAM Journal on Computing*, 40(3):646–663, 2011.

Alberto Rodríguez-Casal. Set estimation under convexity type assumptions. *Annales de l'Institut Henri Poincare (B) Probability and Statistics*, 43(6):763 – 774, 2007.

Aarti Singh, Clayton Scott, and Robert Nowak. Adaptive Hausdorff estimation of density level sets. *The Annals of Statistics*, 37(5B):2760–2782, 2009.

Gurjeet Singh, Facundo Memoli, Tigran Ishkhanov, Guillermo Sapiro, Gunnar Carlsson, and Dario L Ringach. Topological analysis of population activity in visual cortex. *Journal of Vision*, 8(8), 2008.

Alexandre B. Tsybakov. On nonparametric estimation of density level sets. *The Annals of Statistics*, 25(3):948–969, 1997.

Alexandre B Tsybakov and Vladimir Zaiats. *Introduction to Nonparametric Estimation*, volume 11. Springer, 2009.

Jianzhong Wang. *Geometric Structure of High-Dimensional Data and Dimensionality Reduction*. Springer, 2012.

Shmuel Weinberger. The complexity of some topological inference problems. *Foundations of Computational Mathematics*, 14(6):1277–1285, 2014.

Bin Yu. Assouad, Fano, and Le Cam. In *Festschrift for Lucien Le Cam*, pages 423–435. Springer, New York, 1997.

Afra Zomorodian and Gunnar Carlsson. Computing persistent homology. *Discrete & Computational Geometry*, 33(2):249–274, 2005.

Journal of Machine Learning Research 16 (2015) 3637-3686 Submitted 7/13; Revised 11/14; Published 12/15

Supervised Learning via Euler's Elastica Models

Tong Lin LINTONG@PKU.EDU.CN
Hanlin Xue LINLIE312@GMAIL.COM
Ling Wang LING.WANG.NJ@GMAIL.COM
Bo Huang BOHUANG0321@GMAIL.COM
Hongbin Zha ZHA@CIS.PKU.EDU.CN
Key Laboratory of Machine Perception (Ministry of Education)
School of Electronics Engineering and Computer Science
Peking University, Beijing, 100871, China

Editor: Mikhail Belkin

Abstract

This paper investigates the Euler's elastica (EE) model for high-dimensional supervised learning problems in a function approximation framework. In 1744 Euler introduced the elastica energy for a 2D curve on modeling torsion-free thin elastic rods. Together with its degenerate form of total variation (TV), Euler's elastica has been successfully applied to low-dimensional data processing such as image denoising and image inpainting in the last two decades. Our motivation is to apply Euler's elastica to high-dimensional supervised learning problems. To this end, a supervised learning problem is modeled as an energy functional minimization under a new geometric regularization scheme, where the energy is composed of a squared loss and an elastica penalty. The elastica penalty aims at regularizing the approximated function by heavily penalizing large gradients and high curvature values on all level curves. We take a computational PDE approach to minimize the energy functional. By using variational principles, the energy minimization problem is transformed into an Euler-Lagrange PDE. However, this PDE is usually high-dimensional and can not be directly handled by common low-dimensional solvers. To circumvent this difficulty, we use radial basis functions (RBF) to approximate the target function, which reduces the optimization problem to finding the linear coefficients of these basis functions. Some theoretical properties of this new model, including the existence and uniqueness of solutions and universal consistency, are analyzed. Extensive experiments have demonstrated the effectiveness of the proposed model for binary classification, multi-class classification, and regression tasks.

Keywords: supervised learning, Euler's elastica, total variation, geometric regularization, Euler-Lagrange PDE, function approximation, universal consistency

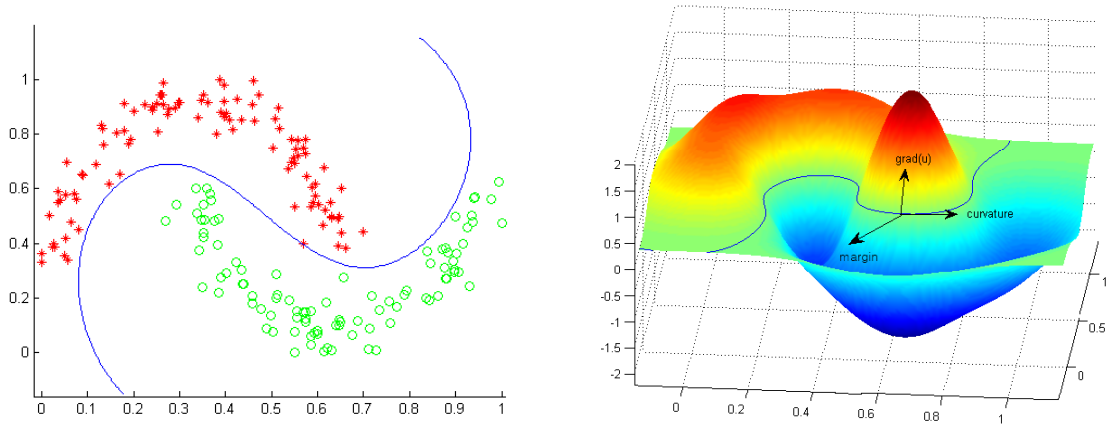
"Read Euler, read Euler, he is our master in everything"
— Pierre-Simon Laplace (1749–1827)

1. Introduction

Supervised learning (Murphy, 2012; Hastie et al., 2009; Bishop, 2006) aims at inferring a function that maps inputs to desired outputs under the guidance of training data. Two main tasks in supervised learning are classification and regression. Numerous supervised learning methods have been developed in several decades; Caruana and Niculescu-Mizil (2006) gave a comprehensive empirical comparison of these methods. A most recent evaluation of classification methods was conducted by Fernández-Delgado et al. (2014): 179 classifiers arising from 17 families were compared on 121 data sets, showing that random forests, support vector machines (SVM), neural networks, and boosting are among the top methods nowadays. Roughly speaking, existing methods can be divided into two main categories: statistics based and function learning based. One advantage of function learning methods is that powerful mathematical theories in functional analysis can be explored rather than doing optimizations on discrete data points.

Most function learning methods can be derived from the energy regularization framework, which minimizes a fitting loss term plus a smoothing penalty. It is arguable that the most successful classification and regression method is the support vector machines (SVM) (Vapnik, 1998; Cristianini and Shawe-Taylor, 2000; Schölkopf and Smola, 2002), whose cost function is composed of a hinge loss and a RKHS norm penalty determined by a kernel. There are several variants of SVM by combining different losses and different penalties (Steinwart, 2005; Bartlett et al., March 2006; Huang et al., 2014). In particular, when replacing the hinge loss by a squared loss, the modified algorithm is called Regularized Least Squares (RLS) method (Rifkin, 2002). Instead of considering a variety of loss terms, manifold regularization (Belkin et al., 2006) introduced a geometric regularizer of squared gradient magnitude on a manifold. Its discrete version corresponds to graph Laplacian regularization (Zhou and Schölkopf, 2005; Nadler et al., 2009). A most recent work is the geometric level set (GLS) classifier (Varshney and Willsky, 2010), with an energy functional composed of a margin-based loss and a geometric regularization term based on the surface area of the decision boundary. The GLS classifier was motivated by the study of minimal surfaces and its applications in image processing. Experiments showed that GLS is competitive with SVM and other state-of-the-art classifiers.

Following the geometric regularization approach, in this paper we propose to use the Euler's elastica for supervised learning problems. The energy functional is composed of a squared loss and an *Euler's elastica* (EE in the sequel) regularizer. Briefly, an elastica regularizer integrates two important geometric factors, gradients and curvatures, in a unified manner. Particularly, its degenerate form is the well-known "total variation" (TV) if only considering gradients and disregarding the influence of curvatures. Since both TV and EE models have achieved great success in image denoising and image inpainting (Chan and Shen, 2005; Aubert and Kornprobst, 2006), a natural question is whether the success of TV and EE models on image processing applications can be transferred to high dimensional data analysis such as supervised learning. This paper investigates the question by extending TV and EE models to supervised learning settings, and evaluating their performance on



(a) (b)

Figure 1: Results on two moon data by using the EE classifier. (a) Decision boundary (in blue) that separates two classes of points (represented by red stars or green circles); (b) learned target function illustrated as a surface in a 3D space.

benchmark data sets against state-of-the-art methods. Figure 1 shows the classification result and the learned target function on the popular example of two moon dataset by using the EE classifier. Note that three important factors considered in the EE classifier, gradient, curvature, and margin between two classes, are depicted in different directions on one data point of the produced decision boundary in Figure 1(b).

Although some researchers in the machine learning community may think that the supervised learning problems have been widely studied and several leading algorithms like SVM (Vapnik, 1998; Cristianini and Shawe-Taylor, 2000; Schölkopf and Smola, 2002), boosting (Schapire and Freund, 2012), and random forests (Breiman, 2001) have been available to achieve superb classification performance, we argue that this work provides a new perspective on understanding supervised learning problems. Particularly, the contributions of this paper are:

1. A proper balance of three important factors in supervised learning: *margin*, *gradient*, and *curvature*. Here the term *margin* refers to the original geometric meaning used in SVM for binary classification problems, namely, the perpendicular distance from a data point to the decision boundary in the input space. The margin of a SVM classifier $\text{sign}(\mathbf{w}\cdot\mathbf{h}(\mathbf{x}))$ can be written as $y(\mathbf{w}\cdot\mathbf{h}(\mathbf{x}))/(\|\mathbf{w}\|_2\|\mathbf{h}(\mathbf{x})\|_2)$, where $\mathbf{w}$ denotes the coefficients of the separating hyperplane, and $\mathbf{h}(\mathbf{x})$ is the high-dimensional feature vector representation of a data point $\mathbf{x}$. Similarly, the margin in boosting can be defined as $y(\mathbf{w}\cdot\mathbf{h}(\mathbf{x}))/(\|\mathbf{w}\|_1\|\mathbf{h}(\mathbf{x})\|_\infty)$, or simply $yf(\mathbf{x})$ if the combined classifier $f(\mathbf{x})$ has been properly normalized (see Schapire and Freund, 2012, chap. 5). Large margins play a central role in developing several state-of-the-art classifiers. Following the traditions in image processing, in this work the squared loss $(y-f(\mathbf{x}))^2$ is used for easier derivative calculations on both classification and regression tasks. Note that

the squared loss is equivalent to a margin-based loss $(1-yf(\mathbf{x}))^2$, called the quadratic loss (Bartlett et al., March 2006, table 1), since $y \in \{-1, +1\}$ in binary classifications. On the other hand, the term *gradient* is related to the slope of function values in a continuous setting, while the *curvature* measures the degree to which all the level curves (including the decision boundary) is curved. Both gradients and curvatures are geometric measurements that reflect the complexity of the output classifier. The trade-off between the squared loss and the complexity involving gradients and curvatures in this work is new to the machine learning community.

2. Euler-Lagrange PDEs that characterize the optimal solution for supervised learning problems. Historically, PDEs have been used to describe a wide range of physical phenomena such as sound, heat, fluid flow, electrostatics, electrodynamics, or elasticity. Surprisingly, these seemingly distinct physical phenomena can be unified under a PDE framework, which implies that they are essentially governed by same or similar nature's mechanism. A natural question is, can PDEs be applicable to high-dimensional supervised learning problems? To the best of our knowledge, Varshney and Willsky (2010) were the first attempt to propose level set based PDEs for classification. Following this research line, we propose the Euler-Lagrange PDEs derived from Euler's elastica model and its degenerate total-variation model, for classification and regression. These PDEs reveal equilibrium conditions of the desired fitting process for supervised learning.

3. Two numerical algorithms for solving the elastica based supervised learning problem in high dimensions. By using radial basis function approximation, we present two PDE solvers: the gradient descent time marching method and the lagged linear equation iteration method.

The remainder of this paper is organized as follows. In Section 2 we begin with a brief review of TV and EE models used in image processing. The proposed models for supervised learning are described in Section 3, followed by the corresponding numerical solutions presented in Section 4. Some theoretical properties of the proposed models are discussed in Section 5. Section 6 presents the experimental results, and Section 7 concludes the paper.

2. Preliminaries

For better understanding the proposed method, we firstly review the notions of total variation and Euler's elastica from an image processing perspective, and point out some connections with prior work in the machine learning literature.

2.1 Total Variation (TV)

A function is said to have bounded variation (BV functions in the sequel) if its total variation is finite. For simplicity we begin with the classical definition of total variation (TV) for a function of one real variable. The total variation of a real-valued function f defined on an

interval $[a, b] \in \mathbb{R}$ is the quantity

$$V_b^a(f) = \sup_P \sum_{i=0}^{n_P-1} |f(x_{i+1}) - f(x_i)|, \tag{1}$$

where the supremum runs over the set of all partitions P of the given interval $[a, b]$, with n_P being the number of points in a specific partition P. If f is differentiable and its derivative is Riemann-integrable, the total variation can be written as

$$V_b^a(f) = \int_a^b |f'(x)|dx.$$

Intuitively it measures the total distance along the direction of the y-axis, neglecting the contribution of motion along x-axis, traveled by a point moving along the graph. Notice that if $f'(x) > 0$ for all $x \in [a, b]$, it is simply equal to $f(b) - f(a)$ by the fundamental theorem of calculus.

The modern definition is based on the concept of distributional derivatives. Let $\Omega \subset \mathbb{R}$ be a bounded open interval. A function $f \in L^1(\Omega)$ is said to be of *bounded variation* (BV) if

$$\sup_\varphi \Big\{ \int_\Omega f(x)\varphi'(x)dx : \varphi \in C_c^1(\Omega), \|\varphi\|_{L^\infty(\Omega)} < 1 \Big\} < \infty, \tag{2}$$

where $C_c^1(\Omega)$ is the space of continuously differentiable functions with compact support in Ω, and $\|\cdot\|_{L^\infty(\Omega)}$ is the essential supremum norm. Note that this definition may have some variants, e.g. imposing the test function that satisfies $\varphi \in C_c^\infty(\Omega)$ and $\|\varphi\|_{C^0(\Omega)} < 1$ (Golubov and Vitushkin, 2001). An equivalent definition is that BV functions are functions whose distributional derivative is a finite Radon measure. Also the two definitions (1) and (2) are consistent. It is natural to generalize the definition (2) for functions of several variables. For an open $\Omega \subset R^d$, the total variation of $f \in L^1(\Omega)$ is given by

$$\sup_\varphi \Big\{ \int_\Omega f\nabla \cdot \varphi \, dx : \varphi = (\varphi_1, \varphi_2, \cdots, \varphi_d) \in C_c^1(\Omega, R^d), \|\varphi\|_{L^\infty(\Omega)} < 1 \Big\} < \infty, \tag{3}$$

where φ is a vector-valued test function, $\nabla \cdot \varphi = \sum \partial\varphi_i/\partial x_i$ is the divergence operator, and all the components of φ has a $L^\infty(\Omega)$-norm less than one. For more details of TV definitions and the BV function space, one can refer to Chan and Shen (2005), Aubert and Kornprobst (2006), Ambrosio et al. (2000), Giusti (1994), and Golubov and Vitushkin (2001).

By penalizing large gradients of the target functions, total variation has been widely used for image processing tasks such as denoising and inpainting. The pioneering work is Rudin, Osher, and Fatemi's image denoising model (Rudin et al., 1992):

$$E[u] = \int_\Omega \Big((u - u_0)^2 + \lambda |\nabla u| \Big) dx,$$

where u_0 is the input image with noise, u is the desired output image, λ is a regulation parameter that balances the two terms, ∇u is the gradient vector $(\partial u/\partial x, \partial u/\partial y)$ for a function $u(x, y)$, $|\nabla u|$ is the l_2-length of the gradient vector, and Ω denotes a $2D$ rectangular image domain. The first fitting term measures the fidelity to the input, while the second

is a p-Sobolev regularization term ($p = 1$) where the gradient ∇u is understood in the distributional sense. The main benefit is to preserve significant image edges during the denoising procedure (Chan and Shen, 2005; Aubert and Kornprobst, 2006), as image edges are important features that should be faithfully retained in image processing. The common downside of TV-based methods is that piecewise constant images with $|\nabla u| = 0$ almost everywhere are favored over piecewise smooth images, which is the so-called staircasing effect (Duan et al., 2013). Euler's elastica model is one of high order approaches to overcome this drawback, which is described in the next subsection.

In the machine learning literature, p-Sobolev regularizer can be found in the literature of nonparametric smoothing splines, generalized additive models, and projection pursuit regression models (Hastie et al., 2009). Specifically, Belkin et al. (2006) proposed the manifold regularization term

$$\int_{x \in M} |\nabla_M u|^2 dx,$$

for any smooth function $u(x)$ on a manifold M. On the other hand, discrete graph Laplacian regularization was discussed in Zhou and Schölkopf (2005) as

$$\sum_{v \in V} |\nabla_v u|^p,$$

where v is a vertex from a vertex set V, and p is an arbitrary number. This penalty measures the roughness of the discrete function u over a graph.

2.2 Euler's Elastica (EE)

The elastica energy first appeared in Euler's work in 1744 on modeling torsion-free thin elastic rods (for the history see Levien, 2008; Fraser, 1991). Then Mumford (1994) reintroduced elastica into computer vision for measuring the quality of interpolating curves in disocclusion. Later, elastica based image inpainting methods were developed in Masnou and Morel (1998) and Chan et al. (2002).

A smooth curve γ is said to be Euler's elastica if it is the equilibrium curve of the elasticity energy:

$$E[\gamma] = \int_\gamma (a + b\kappa^2) ds, \tag{4}$$

where a and b are two non-negative constant weights, κ denotes the scalar curvature (see Appendix A for its definition), and ds is an infinitesimal arc length element. Euler obtained the energy in studying the steady shape of a thin and torsion-free rod under external forces. The curve implies the lowest elastica energy, thus getting its name. The ratio a/b (if $b \neq 0$) indicates the relative importance of the total length versus total squared curvature (Chan and Shen, 2005, chap. 2.1).

According to Mumford (1994), the key link between the elastica curves and image inpainting relies on the the interpolation capability of elasticas. Elasticas were discovered to comply to the *connectivity principle* (Chan and Shen, 2001; Kanizsa, 1979) in visual perception better than total variation. This principle in vision psychology shows that humans mostly prefer having two disjoint parts occluded by another object connected psychologically, even when they are far apart. Such kinds of "nonlinear splines", like classical polynomial splines, are natural tools for completing the missing or occluded edges. Besides, there

is an interesting Bayesian rationale revealed by Mumford (1994) (see also Chan et al., 2002) by considering the *random walk* of a drunk. Suppose the drunk starts from the origin of a 2-D ground and each step is straight. With some distribution assumptions on the step size and the orientation of each step, the maximum likelihood estimation (MLE) of such discrete random walk is approximately equivalent to the minimization of the elastica energy (4) in a continuous fashion. This drunk walking model also sheds light on the choice of "2" for the curvature power in (4). For any $p > 1$, one could consider the general p-elastica energy

$$E_p[\gamma] = \int_\gamma (a + b|\kappa|^p) ds.$$

Notice that the situation of $p = 1$ is less ideal since in this case the total curvature energy permits sudden turns. Chan et al. (2002) pointed out that generic stationary points of the p-elastica energy are forbidden when $p \geq 3$, implying that $p \in (1, 3)$ sounds to be a good choice.

A common approach to bridge the gap between a prior energy model for curves and that for images is using level sets (or called isophotes), pioneered by Osher and Sethian (1988). By "lifting" a curve prior model into a 2D space, one can construct an image prior model imposed on all the level curves of an image (corresponding to a 2D function). Formally, the Euler's elastica of all the level curves of an image u can be expressed as

$$E[u] = \int_{l=0}^{L} \int_{\gamma_l : u=l} (a + b\kappa^2) ds dl, \tag{5}$$

where γ_l is the level curve determined by $u(x) = l$, and the level value l varies in the image range $[0, L]$. Let dt denote an infinitesimal length element along the normal direction $\mathbf{n}$ of the level curve (or along the steepest ascent curve), then we have

$$\frac{dl}{dt} = |\nabla u| \quad \text{or} \quad dl = |\nabla u| dt.$$

Thus by the *co-area formula* (Giusti, 1994), the integrated elastica energy (5) now passes on to u by

$$E[u] = \int_{l=0}^{L} \int_{\gamma_l : u=l} (a + b\kappa^2) |\nabla u| dt ds = \int_\Omega (a + b\kappa^2) |\nabla u| dx,$$

since dt and ds represent a couple of orthogonal length elements. Here Ω denotes the whole rectangular image domain. Now the elastica energy of an image is completely expressed in terms of u, when considering the well known *curvature formula* (Morel and Solimini, 1995) for any level curve $\gamma_l : u(x) = l$

$$\kappa = \nabla \cdot \mathbf{N} = \nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right), \tag{6}$$

where $\nabla \cdot$ denotes the divergence operator, defined as

$$\nabla \cdot \mathbf{V} \doteq \frac{\partial A}{\partial x} + \frac{\partial B}{\partial y}$$

for a vector $\mathbf{V} = (A, B)$, and $\mathbf{N}$ is the ascending unit normal field $\nabla u/|\nabla u|$. See Appendix A for a short derivation of (6). Of course this curvature expression makes sense only for a certain class of smooth functions (such as $C^2(\Omega)$) and requires to be relaxed in order to handle more general functions (like BV or L^1 functions).

Given a small image region D to be inpainted in the whole image domain Ω, Chan and Shen (2005) proposed an inpainting model based on Euler's elastica

$$E = \int_{\Omega \setminus D} (u - u_0)^2 dx + \lambda \int_{\Omega} (a + b\kappa^2)|\nabla u| dx, \tag{7}$$

where λ is a trade-off parameter that balances the first fitting term and the second smoothing term. Notice that the second term in (7) is an elastica regularizer that penalizes high elastica energy on all the level curves of $u(x)$, as expressed in (5). By using *calculus of variation* (van Brunt, 2004), its minimization is reduced to a nonlinear Euler-Lagrange equation. Its numerical method can be implemented by a finite difference scheme, and experimental results show that this elastica based inpainting method performs better than TV based approaches.

Note that total variation can be regarded as a degenerate form of Euler's elastica if setting $a = 1$ and $b = 0$ in (7). In fact, elastica is a combination of total variation that suppresses oscillations in the gradient direction, and a curvature regularizer that penalizes non-smooth level set curves (see Figure 1).

3. The Proposed Framework

We first set up the supervised learning problem, and then introduce three models, Laplacian, total variation, and Euler's elastica, in an increasing order of computational complexity.

3.1 Problem Setup

The general supervised learning problem can be described as follows:

- Given a training data set $\{(\mathbf{x}_1, y_1), ...(\mathbf{x}_n, y_n)\}$ where each data point $\mathbf{x}_i \in \Omega \subset \mathbb{R}^d$ is a d-dimensional column vector and y_i is the corresponding target variable, the goal is to estimate an unknown function $u(\mathbf{x})$ for predicting the desired y on a newly coming point $\mathbf{x}$.

The difference between classification and regression lies only in the corresponding target values, with one discrete and the other continuous. For regression, we simply use $u(\mathbf{x})$ to approximate the target values; for binary classification, the decision boundaries are given by the zero level set of $u(\mathbf{x})$, or $\text{sign}(u(\mathbf{x}))$. Most popular multi-class classifiers are based on some types of reductions to binary classifications; we defer the discussion of multi-class problems to the experiments section.

The widely used functional regularization framework for supervised learning can be formulated as:

$$\min_u \lambda S(u) + \sum_{i=1}^{n} L(y_i, u(\mathbf{x}_i)), \tag{8}$$

where $S(u)$ is a smoothing term or called a penalty and $L(\cdot)$ denotes a loss function. The penalty term is used to control the complexity of the learned function, which has proven to be essential in *Statistical Learning Theory* (Vapnik, 1998; Bousquet et al., 2004; Boucheron et al., 2005; von Luxburg and Schölkopf, 2008). The misclassification risk corresponds to the use of 0-1 loss: $L_{0-1}(y, u(\mathbf{x})) = \mathbf{1}[y \neq \text{sign } u(\mathbf{x})]$, where $\mathbf{1}[\alpha]$ denotes an indicator function that is 1 if α holds true and 0 otherwise. Or we can slightly misuse the notation to allow a margin based representation: $L_{0-1}(y, u(\mathbf{x})) = \mathbf{1}(yu(\mathbf{x}))$, where $\mathbf{1}(\alpha)$ is 1 if $\alpha \leq 0$ and 0 otherwise. It is well known that directly minimizing the 0-1 loss is computationally intractable for many nontrivial classes of functions, and often some nonnegative convex nondecreasing loss function are considered for computational efficiency. Another advantage of such convex surrogates for 0-1 loss is that it is possible to demonstrate the Bayes-risk consistency and to obtain uniform upper bounds on the generalization risk. See Bartlett et al. (March 2006) and Boucheron et al. (2005, Section 4.2) for more discussions.

In the literature a variety of convex surrogate loss functions $L(.)$ have been proposed for binary classification where $y \in \{-1, +1\}$, such as:

1. hinge loss $L_{hinge}(y, u(\mathbf{x})) = \max\{0, 1 - yu(\mathbf{x})\}$ for SVM;
2. squared loss $L_{squared}(y, u(\mathbf{x})) = (y - u(\mathbf{x}))^2$ for RLS;
3. logistic loss $L_{logistic}(y, u(\mathbf{x})) = \log(1 + \exp(-yu(\mathbf{x})))$ for logistic regression;
4. and exponential loss $L_{exponential}(y, u(\mathbf{x})) = \exp(-yu(\mathbf{x}))$ in boosting.

Except for the squared loss, other above losses are margin-based since the classification margin $yu(\mathbf{x})$ is explicitly used. When restricting the discussion on binary classification where $y \in \{-1, +1\}$, the squared loss is actually equivalent to the quadratic loss $(1-yu(\mathbf{x}))^2$ which is then margin-based.

Throughout the paper, the squared loss is used in all our models due to several reasons: (1) The derivative of a squared loss is very simple to calculate; (2) It can be applied to both classification and regression, without any modification; (3) For classification, Rifkin (2002) showed that the RLS method based on squared loss can offer comparable or slightly better accuracies than hinge loss based SVM; (4) Using squared loss is consistent to the related work in image processing area, leading to identical or similar PDEs; (5) We have no intention to exhaustively try and compare different loss functions; instead our focus is on the second term which is a new geometric regularization for supervised learning. For more loss functions and penalties, one can refer to Steinwart (2005), Bartlett et al. (March 2006), and Huang et al. (2014).

Our goal is to explore how TV and EE can be applied to classification and regression problems on high dimensional data sets. To this end, we prefer a continuous integral form rather than the discrete summation form in (8). In contrast to discrete methods such as SVM and graph Laplacian, the proposed framework operates in a continuous fashion where powerful mathematical analysis tools can play a role. Specifically, the calculus of variations plays a role in minimizing the energy functional, leading to the Euler-Lagrange PDE. A typical procedure of this computational PDE approach has three steps: (1) Set up the function learning problem under a continuous setting by designing a proper energy functional; (2) Derive the Euler-Lagrange PDE via the calculus of variations; (3) Finally solve the PDE numerically on discrete data points.

3.2 Laplacian Regularization (LR)

A commonly used model with squared loss can be written as

$$\min_u \lambda S(u) + \sum_{i=1}^{n} \Big(u(\mathbf{x}_i) - y_i\Big)^2.$$

If the RKHS norm is used as the smoothing term $S(u)$, the model is called regularized least squares (RLS) (Rifkin, 2002). Another natural choice is the squared L_2-norm of the gradient: $S(u) = |\nabla u|^2$, as proposed in Belkin et al. (2006). We need to move from the discrete cost function to a continuous functional to leverage powerful mathematical tools. Suppose $\Omega \in \mathbb{R}^d$ is a regular region that contains all the given data points. Under a continuous setting, we have the following Laplacian regularization (LR) model:

$$E_{LR}[u] = \int_\Omega \Big(\lambda |\nabla u|^2 + (u-y)^2\Big) d\mathbf{x}. \tag{9}$$

This LR model has been widely used in the image processing literatures. By calculus of variations (see Appendix B), the minimization is reduced to the following Euler-Lagrange PDE with a *natural boundary condition* over the boundary $\partial\Omega$:

$$\begin{cases} -\lambda \Delta u + (u-y) = 0, \\ \frac{\partial u}{\partial \mathbf{n}}|_{\partial\Omega} = 0, \end{cases} \tag{10}$$

where Δu is the Laplacian operator of u defined as

$$\Delta u \doteq \nabla^2 u = \nabla \cdot \nabla u = \sum_{i=1}^{d} \frac{\partial^2 u}{\partial (x^{(i)})^2},$$

and $\mathbf{n}$ denotes the outer normal of $\partial\Omega$. This PDE (10) is relatively simple and can be easily solved using common methods in two and three dimensions. The next section provides a function approximation method for solving the PDE in high dimensions.

One can observe that the PDE (10) is very similar to the Poisson's equation $-\Delta u = f$ in mathematical physics, where f is a given function. Hence its behavior shares certain degrees of similarity with Poisson's equation. Particularly, if u fits y perfectly (satisfying $u - y = 0$) in a small neighborhood of a particular point $\mathbf{x}$, then by (10) we have $\Delta u = 0$ and further by $u - y = 0$ we also have $\Delta y = 0$ in this neighborhood. On the contrary, if $\Delta y \neq 0$ (implying that $y(\mathbf{x})$ is not a harmonic function), then we can not obtain $u - y = 0$; otherwise by (10) we have $\Delta u = 0$ and $\Delta y = 0$, which is contradictive to our assumption $\Delta y \neq 0$. Therefore, the smoothness of the target variable $y(\mathbf{x})$ determines the fitting degree for supervised learning. The regularization parameter λ controls the strength of this connection.

Throughout the paper, the *natural boundary condition* is adopted for easier treatments. It is well known that boundary conditions can play a significant role in traditional low-dimensional PDE areas, where the shape of the domain boundary is explicitly determined. In these situations, boundary conditions are given by the underlying real problems and their physical meanings are clear. However, in our case of high dimensional spaces for supervised learning, there is no need to specify the exact domain boundary as long as this

domain contains all the data points. Often the input data is preprocessed by scaling each attribute into the range $[-1,+1]$ or $[0,1]$, and hence in practice we define the domain of our TV/EE models as a d-dimensional hypercube. Scaling has been a very important step for using neural networks and SVM, with some advantages discussed in Hsu et al. (2007). Most of these considerations also apply to our algorithms. Recall that our focus is to learn the target function $u(\mathbf{x})$ on an "active" region that contains both the given training data and the future test data, whereas this active region is usually far away from the boundary of the hypercube domain in our settings. Hence boundary values in our high dimensional models are not so important as in low dimensional spaces, and we use the natural boundary condition purely from a computational aspect, just like the related work in image processing. Note that in the GLS classifier (Varshney and Willsky, 2010), the issue of PDE boundary conditions was treated in a similar way.

3.3 Total Variation (TV)

Similar to image denoising, the total variation (TV) model for supervised learning can be formulated as

$$E_{TV}[u] = \int_{\Omega} \Big(\lambda |\nabla u| + \frac{1}{2}(u-y)^2 \Big) d\mathbf{x}. \tag{11}$$

The only difference between LR and TV is just on the p-Sobolev regularizer with $p=2$ for LR and $p=1$ for TV, respectively. Intuitively, LR penalizes gradients on edges too much due to the squared gradient magnitude, while TV is rather milder to permit sharper edges near the decision boundaries between two classes. Similarly, by calculus of variations (see Appendix B) we get the following PDE, which is the exactly same to that in image denoising area:

$$-\lambda \nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) + (u-y) = 0. \tag{12}$$

Note that by the same curvature notation (6) of the associated level hypersurfaces, (12) can be compactly written as

$$-\lambda \kappa + (u-y) = 0. \tag{13}$$

See Appendix A for this curvature notation in R^d, which amounts to the mean curvature up to a constant factor $1/(d-1)$. The PDE (13) implies that the mean curvature κ of all level hypersurfaces with respect to the approximation function $u(\mathbf{x})$ imposes an equilibrium condition on the fitting process of $u-y=0$.

3.4 Euler's Elastica (EE)

The more complicated elastica model for supervised learning can be formulated as

$$E_{EE}[u] = \int_{\Omega} \Big(\lambda (a + b\kappa^2) |\nabla u| + \frac{1}{2}(u-y)^2 \Big) d\mathbf{x}, \tag{14}$$

where κ is given by (6). Due to the elastica regularizer, the final decision boundary and all level sets of $u(\mathbf{x})$ should have a low elastica energy. If setting $a=1$ and $b=0$, this model degenerates to the total variance model. Therefore, a unified solution can be implemented for both TV and EE models, as described in the next section.

Using calculus of variations, we obtain the following PDE for the elastica model:

$$-\lambda \nabla \cdot \mathbf{V}(u) + (u - y) = 0, \tag{15}$$

where the vector field $\mathbf{V}(u)$ is called the *flux* of the elastica energy related to $u(\mathbf{x})$ and can be expressed as a decomposition in a natural orthogonal frame $(\mathbf{N}, \mathbf{T})$:

$$\begin{aligned} \mathbf{V}(u) &\doteq f(\kappa)\mathbf{N} - \frac{\mathbf{T}}{|\nabla u|}\frac{\partial (f'(\kappa)|\nabla u|)}{\partial \mathbf{T}} \\ &= f(\kappa)\mathbf{N} - \frac{1}{|\nabla u|}\Big\{\nabla (f'(\kappa)|\nabla u|) - \mathbf{N}\langle \mathbf{N}, \nabla (f'(\kappa)|\nabla u|)\rangle\Big\} \\ &= f(\kappa)\mathbf{N} - \frac{1}{|\nabla u|}\nabla (f'(\kappa)|\nabla u|) + \frac{1}{|\nabla u|^3}\nabla u\langle \nabla u, \nabla (f'(\kappa)|\nabla u|)\rangle. \end{aligned} \tag{16}$$

Here $f(\kappa) \doteq 1 + b\kappa^2$ by fixing $a = 1$ for simplicity, and $\mathbf{N}$, $\mathbf{T}$ are the normal and tangent vectors given by:

$$\mathbf{N} = \frac{\nabla u}{|\nabla u|}, \quad \mathbf{T} = \mathbf{N}^{\perp}.$$

The directional derivative along $\mathbf{T}$ for a function u is defined as the inner product of ∇u and $\mathbf{T}$:

$$\partial u/\partial \mathbf{T} \doteq \nabla u \cdot \mathbf{T} = \langle \nabla u, \mathbf{T}\rangle.$$

See Appendix B for the detailed derivations from (14) to (15), which originates from Chan et al. (2002). When $b = 0$, (15) degenerates to (12) as $f'(\kappa) = 0$ and $\kappa = \nabla \cdot \mathbf{N}$. Again, the PDE (15) indicates that the divergence of the flux vector field, namely the first term $\nabla \cdot \mathbf{V}(u)$, imposes an equilibrium condition on the fitting process of $u - y = 0$.

4. Numerical Algorithms

Due to the nonlinearity of the regularizer in TV and EE models, the corresponding PDEs in (12) and (15) are too complicated to be efficiently solved in high dimensional space. Even though the PDE in (10) associated with the LR model can be solved by Finite Difference Method (FDM) or Finite Element Method (FEM) in 2-D or 3-D spaces, currently we have no PDE tools to deal with such high dimensional problems. Therefore we take a function approximation idea by using radial basis functions (RBF), similar to the treatment in GLS (Varshney and Willsky, 2010). Then the computational PDE problems can be reduced to finding the expanding coefficients.

In the literature of image denoising and inpainting, dynamic programming was firstly employed to solve elastica related image processing problems in Masnou and Morel (1998). The most widely used method is the computational PDE approach (Chan and Shen, 2005; Aubert and Kornprobst, 2006), partially due to the following reasons:

1. The theory of PDEs is well established;

2. Many variational problems or their regularized approximations can often be effectively computed from their Euler-Lagrange equations;

3. As in classical mathematical physics, PDEs are powerful tools to describe, model, and simulate many dynamic as well as equilibrium phenomena.

Later in Bae et al. (2011) and Komodakis and Paragios (2009), graph-cuts methods are applied to elastica models. Several numerical solutions (Tai et al., 2011; Hahn et al., 2011; Duan et al., 2013) are based on the operator splitting technique and the augmented Lagrangian method (ALM), which decomposes the original problem into a series of subproblems. All subproblems are either linear which can be solved efficiently by iterative solvers, or having closed-form solutions. Recently, Bredies et al. (2013) proposed a convex, lower semi-continuous approximation of Euler's elastica energy on image processing tasks via functional lifting, which can be expressed as a linear program. However, it is still unclear whether these newly developed numerical methods are applicable to high dimensional elastica problems.

4.1 Approximation by Radial Basis Functions

The function approximation idea relies on the fact that a function $u(\mathbf{x})$ can be expressed as a sum of weighted basis function $\{\phi_i(\mathbf{x})\}$. For instance, a Taylor expansion represents a function by using polynomials as basis functions. The Ritz method is a direct method for solving problems in variational calculus by means of a linear combination of known basis functions. In the literature of machine learning, the most widely used are the Gaussian radial basis function (RBF) kernels, which are simple in expressions but have powerful fitting ability. Hence we assume that the function $u(\mathbf{x})$ to be learned has the following representation

$$u(\mathbf{x}) = \sum_{i=1}^{n} w_i \phi_i(\mathbf{x}), \tag{17}$$

where $\{\phi_i(\mathbf{x})\}$ are a set of Gaussian RBF kernels

$$\phi_i(\mathbf{x}) \doteq \exp(-\frac{1}{2} c||\mathbf{x} - \mathbf{x}_i||^2).$$

Here $\{\mathbf{x}_i\}$ are the training samples in supervised learning, and c is a tunable parameter. Note that the granularity of this representation is well-matched to the data size, as the number of RBFs is equal to the number of training samples. By using the RBF approximation, the problem is reduced to finding the coefficients $\{w_i\}$. Hence our approach is similar to kernel machines with the Gaussian RBF kernels since the decision function is formulated as a linear combination of RBFs. The main difference is that our approach is based on the Euler's elastica regularization term, while kernel methods in the literature employs a squared norm of reproducing kernel Hilbert space for regularization.

Though there are numerous basis functions (also known as kernels) being proposed by researchers, four basic types are often considered in the SVM literature and related books: linear, polynomial, sigmoid, and Gaussian RBFs. In Hsu et al. (2007) the Gaussian RBF kernel is suggested to be a reasonable first choice for training SVMs due to several reasons. Most of these considerations also apply to our algorithms, such as the number of hyperparameters, and the difficulties in numerical computations. In addition, one might consider other types of RBFs instead of Gaussians, like compactly supported RBFs used in scattered data interpolation (Wendland, 1995; Floater and Iske, 1996). The main purpose of

compactly supported RBFs is for reducing computational complexity. However, the usage of compactly supported RBFs might lead to numerical difficulties in the following derivative calculations in our algorithms.

Let $\mathbf{H}(u)$ denote the Hessian matrix of u, and $\mathbf{I}$ be an identity matrix with a proper size. For short notations we also use ϕ_i for $\phi_i(\mathbf{x})$. Based on the RBF approximation (17), the following are some analytical expressions and handy notations that will be frequently used later. See Appendix C for some derivations of these expressions. Note that d is the dimension of the feature space.

$$\begin{aligned}
\nabla\phi_i &= -c(\mathbf{x}-\mathbf{x}_i)\phi_i, \\
\Delta\phi_i &= c(c|\mathbf{x}-\mathbf{x}_i|^2 - d)\phi_i, && (18)\\
\mathbf{H}(\phi_i) &= -c\phi_i\mathbf{I} + c^2(\mathbf{x}-\mathbf{x}_i)(\mathbf{x}-\mathbf{x}_i)^T\phi_i, && (19)\\
\nabla u &= \sum_i w_i\nabla\phi_i = -c\sum_i w_i(\mathbf{x}-\mathbf{x}_i)\phi_i = -c\mathbf{g}, \\
\mathbf{g} &\doteq \sum_i w_i(\mathbf{x}-\mathbf{x}_i)\phi_i, && (20)\\
\Delta u &= \sum_i w_i\Delta\phi_i = c\sum_i w_i(c|\mathbf{x}-\mathbf{x}_i|^2 - d)\phi_i, \\
\mathbf{H}(u) &= -c\Big(\sum_i w_i\phi_i\Big)\mathbf{I} + c^2\Phi, && (21)\\
\Phi &\doteq \sum_i w_i(\mathbf{x}-\mathbf{x}_i)(\mathbf{x}-\mathbf{x}_i)^T\phi_i, \\
\mathbf{N} &\doteq \frac{\nabla u}{|\nabla u|} = -\frac{\mathbf{g}}{|\mathbf{g}|}, \\
\kappa &\doteq \nabla\cdot\frac{\nabla u}{|\nabla u|} && (22)\\
&= \frac{1}{|\nabla u|}\left(\Delta u - \frac{\nabla u^T H(u)\nabla u}{\nabla u^T\nabla u}\right) \\
&= \frac{1}{|\mathbf{g}|}\Big\{\sum_i w_i(c|\mathbf{x}-\mathbf{x}_i|^2 - d + 1)\phi_i - c\frac{\mathbf{g}^T\Phi\mathbf{g}}{\mathbf{g}^T\mathbf{g}}\Big\}. && (23)
\end{aligned}$$

4.2 Algorithm for LR

First, let us consider how to deal with the simplest LR model by solving the linear elliptic PDE (10): $-\lambda\Delta u + (u-y) = 0$. By using the RBF approximation (17) and the linearity of the Laplacian operator, the goal is reduced to finding a set of weights $\{w_i\}$:

$$\sum_i w_i(\phi_i - \lambda\Delta\phi_i) = y.$$

Let $\mathbf{w} \doteq (w_1, w_2, ..., w_n)^T$ and $\mathbf{y} \doteq (y_1, y_2, ..., y_n)^T$, where n is the number of training samples. Then $\mathbf{w}$ can be solved by the system of linear equations:

$$\mathbf{A}\mathbf{w} = \mathbf{y}, \;\; \mathbf{A}_{ij} = \phi_j(\mathbf{x}_i) - \lambda\Delta\phi_j(\mathbf{x}_i).$$

Numerically, the following regularized least squares solution is adopted in practice to avoid ill-posed problems:

$$\min_{\mathbf{w}} |\mathbf{A}\mathbf{w} - \mathbf{y}|^2 + \eta|\mathbf{w}|^2.$$

The closed-form solution is simply given by $\mathbf{w} = (\mathbf{A}^T\mathbf{A} + \eta\mathbf{I})^{-1}\mathbf{A}^T\mathbf{y}$ with fast computational speed. It is interesting to see that both classification and regression problems can be solved by fitting a set of linear equations. Naturally, the LR method can be regarded as a generalization of linear regression $\mathbf{X}\mathbf{w} = \mathbf{y}$ or ridge regression $\min_{\mathbf{w}} |\mathbf{X}\mathbf{w} - \mathbf{y}|^2 + \eta|\mathbf{w}|^2$ (Hastie et al., 2009, chap. 3), where the original data matrix $\mathbf{X}$ is replaced by a "new" data matrix $\mathbf{A}(\mathbf{X})$ in the LR model.

4.3 Algorithm for TV and EE Models

As the TV model is one degenerate case of the EE model, we describe solutions for the more complicated EE model in this section. Here two algorithms are developed to tackle the nonlinearity in (15): (1) gradient descent time marching, and (2) lagged linear equation iteration.

4.3.1 Gradient Descent Time Marching

A standard solution is the steepest gradient descent marching with an artificial time t:

$$\frac{\partial u(\mathbf{x}, t)}{\partial t} = -\frac{\partial E_{TV}}{\partial u} = \lambda\nabla \cdot \Big(\frac{\nabla u}{|\nabla u|}\Big) - (u - y) \tag{24}$$

for the total variation PDE (12) and

$$\frac{\partial u(\mathbf{x}, t)}{\partial t} = -\frac{\partial E_{EE}}{\partial u} = \lambda\nabla \cdot \mathbf{V} - (u - y) \tag{25}$$

for the elastica PDE (15). Note that by setting $u_t = -E_u$, the energy functional E should decrease in the gradient direction as time marching. Here the partial derivative E_u can be obtained from the first variation of E (see Appendix A).

For image processing tasks, these gradient descent flows can be processed on a natural regular grid of the image domain. For high dimensional data space, such computational process is prohibitive. With the function approximation (17), a more practical way is handling the gradient descent flow about the weight vector $\mathbf{w}$. Consider a matrix form of the function approximation (17) on all training data points:

$$\mathbf{u} \doteq \begin{pmatrix} u(\mathbf{x}_1) \\ \vdots \\ u(\mathbf{x}_n) \end{pmatrix} = \Psi\mathbf{w},\ \Psi_{ij} \doteq \phi_j(\mathbf{x}_i).$$

Thus we have the gradient descent flow about $\mathbf{w}$:

$$\frac{\partial \mathbf{w}}{\partial t} = \Psi^{-1}\frac{\partial \mathbf{u}}{\partial t} = \Psi^{-1}\begin{pmatrix} \frac{\partial u}{\partial t}|_{\mathbf{x}=\mathbf{x}_1} \\ \vdots \\ \frac{\partial u}{\partial t}|_{\mathbf{x}=\mathbf{x}_n} \end{pmatrix}.$$

Then in each iteration the weight vector $\mathbf{w}$ can be updated by

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} + \tau \Psi^{-1} \begin{pmatrix} \frac{\partial u^{(k)}}{\partial t}|_{\mathbf{x}=\mathbf{x}_1} \\ \vdots \\ \frac{\partial u^{(k)}}{\partial t}|_{\mathbf{x}=\mathbf{x}_n} \end{pmatrix},$$

where τ is a small time step. We first initialize the weight vector $\mathbf{w}$ as $\mathbf{w}^{(0)} = (\Psi^T\Psi + \eta\mathbf{I})^{-1}\Psi^T\mathbf{y}$ by solving the regularized least squares problem $\Psi\mathbf{w} = \mathbf{y}$, with η a regularization parameter. Then we get $u^{(0)} = \Psi\mathbf{w}^{(0)}$, and run the iteration by computing $\mathbf{w}^{(k+1)}$ and $u^{(k+1)}$ alternately.

Here we give some details about the computation of the partial derivatives. Clearly the partial u_t in (24) can be obtained by (23). By omitting the third and higher order terms, $\nabla \cdot \mathbf{V}$ can be expanded into the following expression (see Appendix D):

$$\nabla \cdot \mathbf{V} = \kappa + b\kappa^3 - \frac{2b(\Delta u)^2}{|\nabla u|^5}\alpha + 6b\Big(\frac{\Delta u}{|\nabla u|^7} - \frac{\kappa}{|\nabla u|^6}\Big)\alpha^2 + \frac{6b}{|\nabla u|^7}\alpha\beta + \frac{2b}{|\nabla u|^5}\gamma, \tag{26}$$

where

$$\alpha \doteq \nabla u^T \mathbf{H}(u)\nabla u, \quad \beta \doteq \nabla u^T \mathbf{H}(u)^2\nabla u, \quad \gamma \doteq \nabla u^T \mathbf{H}(u)^3\nabla u.$$

We can see that if by setting $b = 0$, the expression of $\nabla \cdot \mathbf{V}$ is degenerated to $\kappa = \nabla \cdot (\nabla u/|\nabla u|)$, which is exactly the same expression of the TV model.

The time complexity in each iteration is $O(n^2 d)$, where n is the number of data points and d is the dimension. There are 3 parameters in the algorithm: the RBF parameter c, the regularization parameter λ, and the elastica weight parameter b. Note that we always set $a = 1$ since a can be absorbed into λ.

4.3.2 Lagged Linear Equation Iteration

Following the spirit of the lagged diffusivity fixed-point iteration method (Chan and Shen, 2005), we develop the following lagged linear equation iteration method. Empirically, the original lagged diffusivity fixed-point iteration often yields poor performance due to its brute-force linearization on the nonlinear PDE.

For the simpler TV model, by expanding the curvature term with (23) we have

$$-\frac{\lambda}{|\nabla u|}\Big(\Delta u - \frac{\nabla u^T H(u)\nabla u}{\nabla u^T \nabla u}\Big) + (u - y) = 0,$$

or equivalently by the RBF approximation

$$-\lambda\Big\{\sum_i w_i(1 - d + c|\mathbf{x} - \mathbf{x}_i|^2)\phi_i - c\frac{\mathbf{g}^T\Phi\mathbf{g}}{\mathbf{g}^T\mathbf{g}}\Big\} + |\mathbf{g}|\Big\{\Big(\sum_i w_i\phi_i\Big) - y\Big\} = 0.$$

The above nonlinear equation about $\mathbf{w}$ is rather complex as $\mathbf{g}$ and Φ contain the unknown $\mathbf{w}$. To simplify this equation, we use an iteration method that computes $\mathbf{w}$ or $\mathbf{g}$ alternately by fixing the other variables. First, $\mathbf{w}$ is initialized as a random vector. Then $\mathbf{g}$ can be

computed according to (20). Now assuming that $\mathbf{g}$ is fixed, we have

$$\begin{aligned}\frac{\mathbf{g}^T\Phi\mathbf{g}}{\mathbf{g}^T\mathbf{g}} &= \frac{\mathbf{g}^T[\sum_i w_i(\mathbf{x}-\mathbf{x}_i)(\mathbf{x}-\mathbf{x}_i)^T\phi_i]\mathbf{g}}{\mathbf{g}^T\mathbf{g}} \\ &= \sum_i w_i\phi_i\Big(\frac{\mathbf{g}^T(\mathbf{x}-\mathbf{x}_i)(\mathbf{x}-\mathbf{x}_i)^T\mathbf{g}}{\mathbf{g}^T\mathbf{g}}\Big).\end{aligned}$$

Thus the original nonlinear equation about $\mathbf{w}$ becomes a linear equation

$$\sum_i w_i\Big(\frac{|\mathbf{g}|}{\lambda}-h\Big)\phi_i = \frac{|\mathbf{g}|}{\lambda}y,$$

where

$$h \doteq 1-d+c|\mathbf{x}-\mathbf{x}_i|^2 - c\frac{\mathbf{g}^T(\mathbf{x}-\mathbf{x}_i)(\mathbf{x}-\mathbf{x}_i)^T\mathbf{g}}{\mathbf{g}^T\mathbf{g}}.$$

Using the lagged idea, we obtain the method of lagged linear equation iteration: (1) By fixing $\mathbf{g}$, solve the system of linear equations with respect to $\mathbf{w}$ to get a new $\mathbf{w}$; (2) Compute $\mathbf{g}$ with the updated $\mathbf{w}$; (3) Iterate until convergence or reaching maximal iteration number.

For the more complicated EE model, we have to simplify the corresponding PDE greatly. Following the lagged idea again, we first assume the term about curvature $K \doteq a+b\kappa^2$ being fixed. Then K can be absorbed into λ, leading to the following linear equation in a similar way:

$$\sum_i w_i\Big(\frac{|\mathbf{g}|}{\lambda K}-f\Big)\phi_i = \frac{|\mathbf{g}|}{\lambda K}y.$$

Similarly, a two-step lagged iteration procedure can be developed for the EE model: (1) By fixing $\mathbf{g}$ and K, solve the linear system with respect to $\mathbf{w}$; (2) Compute $\mathbf{g}$ and K with the updated $\mathbf{w}$; (3) Iterate until convergence or reaching maximal iteration number. There are three parameters: c, λ, and the regularization parameter η (empirically chosen in experiments) in the least squares problems.

5. Theoretical Properties

In this section, we explore some theoretical analysis for elastica based supervised learning algorithms under the framework of statistical learning theory (SLT) (Vapnik, 1998; Bousquet et al., 2004; Boucheron et al., 2005; von Luxburg and Schölkopf, 2008). First we present the existence and uniqueness analysis of our TV/EE solutions. Then we prove that elastica based classifiers are universally consistent, mainly based on the pioneering work of Steinwart (2005) for SVM and other regularized kernel classifiers.

5.1 Existence and Uniqueness of TV

We first consider the TV model (11), which is a special yet useful case of the elastica model (14). It is well-known that one can carry out the existence and uniqueness analysis for TV model in image processing tasks. Thanks to the fact that most properties of a BV function are independent of the data dimension, the following proof in $\mathbb{R}^d$ is a trivial but detailed

extension of the overly simplified proof for the TV-based image denoising model in (Chan and Shen, 2005, Theorem 4.14 in chap. 4).

Before giving the theorem on existence and uniqueness, we first review several major properties of BV functions (Chan and Shen, 2005, Section 2.2.2) (Aubert and Kornprobst, 2006, Section 2.2.3) that are frequently used in the following proofs.

Theorem 1 *(1)* (Completeness) $\mathrm{BV}(\Omega) \subset L^1(\Omega)$ *is a Banach space under the BV norm*

$$\|u\|_{BV} \doteq \int_\Omega (|u| + |\nabla u|) d\mathbf{x}.$$

(2) (Weak Compactness) *Let* $\{u_n\}$ *be a bounded sequence in* $\mathrm{BV}(\Omega)$ *where* Ω *is a Lipschitz domain. There must exist a subsequence which converges in* $L^1(\Omega)$.
(3) (L^1-Lower Semicontinuity) *Suppose a sequence* $\{u_n\}$ *converges to* u *in* $L^1(\Omega)$. *Then*

$$\int_\Omega |\nabla u| d\mathbf{x} \leq \liminf_n \int_\Omega |\nabla u_n| d\mathbf{x}.$$

In particular if $\{u_n\}$ *is a bounded sequence in* $\mathrm{BV}(\Omega)$, *then* u *belongs to* $\mathrm{BV}(\Omega)$ *as well.*

Theorem 2 (Existence and Uniqueness of TV) *Under the assumption that the given target function* $y(\mathbf{x}) \in L^2(\Omega)$ *with* $\mathbf{x} \in R^d$, *the minimization problem*

$$E_{TV}[u] = \int_\Omega \Big(\frac{1}{2}(u-y)^2 + \lambda |\nabla u| \Big) d\mathbf{x}$$

admits a unique solution $\hat{u}(\mathbf{x}) \in \mathrm{BV}(\Omega)$.

Proof We first show the existence. E_{TV} is finite for at least one BV function $\bar{u}(\mathbf{x}) \equiv \int_\Omega y(\mathbf{x}) d\mathbf{x}$, which is a constant function over Ω with $|\nabla \bar{u}| = 0$. Thus there exist some BV functions having finite E_{TV} values. Clearly 0 is a lower bound of these E_{TV} values. Hence this nonempty number set of all E_{TV} values with 0 as a lower bound must have an infimum denoted as $E_0 (\geq 0)$. Since E_0 is an infimum, we can select a sequence of BV functions $\{u_i\}$ with bounded E_{TV} values such that their E_{TV} values converges to E_0. Note that such sequence of $\{u_i\}$ must be bounded as well as in $\mathrm{BV}(\Omega)$ in terms of the BV norm, since the TV seminorm $\int_\Omega |\nabla u| d\mathbf{x}$ is contained in E_{TV} and $\mathrm{BV}(\Omega) \subset L^1(\Omega)$. According to the weak compactness of the BV space, for the bounded sequence $\{u_i\}$ in $\mathrm{BV}(\Omega)$, there must exist a subsequence indexed by $i(k), k = 1, 2, \ldots$, which converges in $L^1(\Omega)$. Due to the completeness of $L^1(\Omega)$, let $\hat{u} \in L^1(\Omega)$ be its limit. By the L^1-lower semicontinuity of the TV seminorm, we have

$$\int_\Omega |\nabla \hat{u}| d\mathbf{x} \leq \liminf_k \int_\Omega |\nabla u_{i_k}| d\mathbf{x}$$

and also $\hat{u} \in \mathrm{BV}(\Omega)$ since $\{u_i\}$ is a bounded sequence in $\mathrm{BV}(\Omega)$. Observe that E_{TV} is lower semicontinuous with respect to the $L^1(\Omega)$ topology because both of its components, the L^2 norm (the squared loss in E_{TV}) and the TV seminorm, are lower semicontinuous. That is,

$$E_{TV}[\hat{u}] \leq \liminf_k E_{TV}[u_{i_k}] = \inf_{u \in \mathrm{BV}(\Omega)} E_{TV}[u] = E_0,$$

indicating that there exists $\hat{u} \in \mathrm{BV}(\Omega)$ achieving the minimum point of E_{TV}.

The uniqueness follows directly from the strict convexity of E_{TV}. Thanks to the Minkowski inequality $\|f+g\|_{L^p} \leq \|f\|_{L^p} + \|g\|_{L^p}$, the TV seminorm is convex (but not strictly convex) given by

$$\int_\Omega |\nabla(\alpha u + (1-\alpha)v)| = \int_\Omega |\alpha\nabla u + (1-\alpha)\nabla v| \leq \alpha \int_\Omega |\nabla u| + (1-\alpha)\int_\Omega |\nabla v|,$$

where $\alpha \in [0,1]$. Apparently the L^2 norm $\int_\Omega (u-y)^2$ is strictly convex. Hence combining two components together, we have that E_{TV} is strictly convex. Therefore as the minimum point of E_{TV}, $\hat{u} \in \mathrm{BV}(\Omega)$ is unique. ■

In the image processing literature, there are some variants of the existence and uniqueness analysis for different TV models. Chan et al. (2002) discussed the existence of TV inpainting models in the cases of noise free and having noise, but the uniqueness is neglected. Aubert and Kornprobst (2006) present the existence and uniqueness analysis for the TV-based image restoration problem

$$\min E_{TV}[u] = \int_\Omega \Big(\frac{1}{2}(Ru-y)^2 + \lambda\phi(|\nabla u|)\Big)d\mathbf{x},$$

where R is a linear blurring operator and ϕ is a strictly convex and nondecreasing cost function.

5.2 Existence of EE

We now consider the more complicated elastica model. In Ambrosio and Masnou (2003), the authors proved that a relaxed version of elastica-based image inpainting has at least one solution in $\mathrm{BV}(\Omega)$. Here we give the existence proof of a discrete elastica model for binary classification, which is adapted from the elegant proof in Steinwart (2005) for SVMs and other regularized kernel classifiers. The existence is the first step to fulfill the consistency proof in the next subsection. But the solution to elastica model can be non-unique, due to the lack of convexity for this energy functional.

We begin with some preliminary notations. In the following, let $\overline{\mathbb{R}} = [-\infty, +\infty]$, $\mathbb{R}^+ = [0, +\infty)$, and $\overline{\mathbb{R}}^+ = [0, +\infty]$. A binary classifier is a rule that assigns to every *training set* $T = \{(x_1, y_1), \ldots, (x_n, y_n)\} \in (X \times Y)^n$ ($Y = \{-1, +1\}$ for binary problems) a measurable function $f : X \to \mathbb{R}$ with the final decision given by $\mathrm{sign} f(x)$. Similar to the gray scale constraint in image processing tasks, we assume that f takes values in a bounded interval (e.g. $[-2, 2]$) since f should approximate $y \in \{-1, +1\}$ and the classification decision is only rated with the sign of f. Sometimes we use a looser condition that $f \in L_\infty(X)$. For a given loss function $L(y, f(x))$, write a **cost function** $C(\alpha, t) \doteq \alpha L(1, t) + (1-\alpha)L(-1, t)$ for $\alpha \doteq P(Y = 1|X = x) \in [0, 1]$ and $t \in \overline{\mathbb{R}}$. For a fixed α, define $M(\alpha)$ and the corresponding t_α such that $M(\alpha) \doteq C(\alpha, t_\alpha) \doteq \min_t C(\alpha, t)$. We then give the basic condition on the loss function L in order to guarantee that the solution t_α minimizing $C(\alpha, t)$ tends to have the same sign as the Bayes decision rule.

Definition 3 *A continuous function $L(y, f(x))$ is called an **admissible** loss function if for every $\alpha \in [0, 1]$ and $t_\alpha \in \overline{\mathbb{R}}$ we have $t_\alpha < 0$ if $\alpha < 1/2$ and $t_\alpha > 0$ if $\alpha > 1/2$.*

A similar concept called **classification-calibrated** can be found in Bartlett et al. (March 2006), requiring that an incorrect sign of t_α always leads to a strictly larger $M(\alpha)$. The classification-calibrated condition generalizes the requirement of an admissible loss that the minimizer of $C(\alpha, t)$ (if it exists) has the correct sign. The admissibility of L is necessary in order to develop universally consistent classifiers (Steinwart, 2005). In particular, the quadratic loss $L(y, f(x)) = (1 - yf(x))^2$ used in our classification models is admissible and classification-calibrated; other examples can be found in Steinwart (2005) and Bartlett et al. (March 2006). In the following we always assume that $L(y, f(x))$ is a margin-based admissible loss function which is continuous with respect to the margin $yf(x)$.

Definition 4 *Let $S(\lambda, t) : \mathbb{R}^+ \times \overline{\mathbb{R}}^+ \to \overline{\mathbb{R}}^+$ be an increasing function with respect to λ and t, which is continuous in 0 with respect to λ and unbounded with respect to t. Moreover, for all $\lambda > 0$ there exists a $t > 0$ such that $S(\lambda, t) < \infty$. We call $S(\lambda, t)$ a **regularization function** if for all $\lambda > 0$ and $s \in \mathbb{R}^+$ we have $S(\lambda, 0) = S(0, s) = 0$, and if for all $\lambda > 0$, $t \in \overline{\mathbb{R}}^+$, and for all sequences $\{t_n\} \subset \overline{\mathbb{R}}^+$ with $t_n \to t$ and $S(\lambda, t_n) < \infty$, we have $S(\lambda, t_n) \to S(\lambda, t)$.*

In our TV/EE models, $S(\lambda, t) = \lambda t^2$ clearly satisfies the requirements of a regularization function. This regularization function is a typical setting in several variants of SVMs (Steinwart, 2005), leaving the differences of these variants mainly on the loss functions.

Definition 5 *The **(0-1) risk** of a measurable function $f : X \to \mathbb{R}$ is defined by*

$$\begin{aligned} R_P(f) &\doteq P(\{(x, y) : \mathrm{sign} f(x) \neq y\}) \\ &= \mathbb{E}_{(x,y)\sim P} 1(y\, f(x)). \end{aligned}$$

The smallest achievable risk

$$R_P \doteq \inf\{R_P(f) : f : X \to \mathbb{R} \text{ measurable}\}$$

*is called the **Bayes risk** of P.*

Definition 6 *Given an admissible loss function L and a probability measure P, the **L-risk** of a measurable function $f : X \to \mathbb{R}$ is defined by*

$$\begin{aligned} R_{L,P}(f) &\doteq \mathbb{E}_{(x,y)\sim P} L(y, f(x)) \\ &= \int_{(x,y)\sim P} L(y, f(x)) P_X(dx) P_Y(dy) \\ &= \int_X C(P(Y = 1|X = x), f(x)) P_X(dx). \end{aligned}$$

*The smallest possible L-risk is denoted by $R_{L,P}$. Furthermore, given a regularization function S, the **regularized L-risk** is defined by*

$$R^{reg}_{L,P,\lambda}(f) \doteq S(\lambda, \|f\|_{EE}) + R_{L,P}(f)$$

for all $\lambda > 0$. Here $\|f\|^2_{EE} \doteq \int_X (1 + b\kappa^2)|\nabla f| dx$ is the Euler's elastica regularizer with a misused norm notation, and $\kappa = \nabla \cdot \left(\frac{\nabla f}{|\nabla f|}\right)$. If overlooking the curvature term, it degenerates to the TV seminorm $\|f\|^2_{TV} \doteq \int_X |\nabla f| dx$. If P is an empirical measure with respect to $T \in (X \times Y)^n$, we write $R_{L,T}(f)$ and $R^{reg}_{L,T,\lambda}(f)$, respectively.

Theorem 7 (Existence of EE) *For all Borel probability measures P on $X \times Y$ and all $\lambda > 0$, there always exists a function $f_{P,\lambda} \in \mathrm{BV}(X)$ minimizing the regularized L-risk $R^{reg}_{L,P,\lambda}(f)$. Moreover, for all such $f_{P,\lambda} \in \mathrm{BV}(X)$ we have $\|f_{P,\lambda}\|_{EE} \leq \delta_\lambda$ where*

$$\delta_\lambda \doteq \sup\{t : S(\lambda, t) \leq 2[L(1,0) + L(-1,0)]\}.$$

Proof The following proof is adapted from Steinwart (2005, Lemma 3.1), and the difference lies on $R^{reg}_{L,P,\lambda}(f)$ where the original RKHS norm $\|f\|_H$ for SVM is replaced by the pseudo-norm $\|f\|_{EE}$ for EE. The proof consists of the following five steps.

A. Clearly $R^{reg}_{L,P,\lambda}(f)$ is finite for the BV function $\bar{f}(\mathbf{x}) \equiv \mathbb{E}_{(x,y)\sim P}\, y$ or $\bar{f}(\mathbf{x}) \equiv 0$, which is a constant function over X with $|\nabla \bar{f}| = 0$. Thus there exist some BV functions having finite $R^{reg}_{L,P,\lambda}(f)$ values. For all $\varepsilon \in (0, L(1,0) + L(-1,0)]$, by the definition of an infimum we can select an function $f_\varepsilon \in L^1(X)$ with

$$R^{reg}_{L,P,\lambda}(f_\varepsilon) \leq \inf_{f \in L^1(X)} R^{reg}_{L,P,\lambda}(f) + \varepsilon.$$

Now we have

$$\begin{aligned}
\inf_{f \in L^1(X)} R^{reg}_{L,P,\lambda}(f) &\leq R^{reg}_{L,P,\lambda}(f \equiv 0) \\
&= S(\lambda, \|f \equiv 0\|_{EE}) + R_{L,P}(f \equiv 0) \\
&= 0 + \mathbb{E}_{(x,y)\sim P} L(y, f(x) \equiv 0) \\
&= P(y = 1|x)L(1,0) + P(y = -1|x)L(-1,0) \\
&\leq L(1,0) + L(-1,0),
\end{aligned}$$

where S satisfies the condition $S(\lambda, 0) = 0$ in the second equality. Furthermore,

$$\begin{aligned}
S(\lambda, \|f_\varepsilon\|_{EE}) &\leq S(\lambda, \|f_\varepsilon\|_{EE}) + R_{L,P}(f_\varepsilon) = R^{reg}_{L,P,\lambda}(f_\varepsilon) \\
&\leq \inf_{f \in L^1(X)} R^{reg}_{L,P,\lambda}(f) + \varepsilon \leq 2[L(1,0) + L(-1,0)].
\end{aligned}$$

As $S(\lambda, t)$ is an increasing function with respect to t, we obtain the boundedness of $\|f_\varepsilon\|^2_{EE}$. Since $\|f\|^2_{TV} \leq \|f\|^2_{EE}$, we also have the boundedness of $\|f_\varepsilon\|^2_{TV}$ and $f_\varepsilon \in \mathrm{BV}(X)$.

B. The *Bolzano-Weierstrass theorem* states that each bounded sequence in $\mathbb{R}^n$ has a convergent subsequence. In functional analysis, the *Eberlein-Smulian theorem* (Conway, 1990, Theorem 13.1 in chap. 5) states that three different kinds of weak compactness are equivalent in a Banach space. Particularly, we will use the sequential compactness property of a subset A in a Banach space: *Every sequence from A has a convergent subsequence whose limit is in A in the weak sense.* Recall that $\mathrm{BV}(X)$ is a Banach space. By the two theorems, there exist $f_{P,\lambda} \in \mathrm{BV}(X)$, a sequence $\{f_{\varepsilon_n}\} \in \mathrm{BV}(X)$, and two finite number $c_1, c_2 \in \mathbb{R}^+$ such that $\|f_{\varepsilon_n}\|_{EE} \to c_1$, $\|f_{\varepsilon_n}\|_{TV} \to c_2$, and $f_{\varepsilon_n} \to f_{P,\lambda}$ weakly. Note that the weak convergence implies that $f_{P,\lambda}$ is uniquely determined, $\|f_{P,\lambda}\|_{BV} \leq \liminf_n \|f_{\varepsilon_n}\|_{BV}$, and $\|f_{P,\lambda}\|_{L^1} \leq \liminf_n \|f_{\varepsilon_n}\|_{L^1}$ since $\mathrm{BV}(X) \subset L^1(X)$ (Yosida, 1999, Theorem 5 and 9 in Chapter V.1). In particular, by the weak compactness of the BV space, we further have that $\{f_{\varepsilon_n}\}$ converges to $f_{P,\lambda}$ in $L^1(X)$. Thus $yf_{\varepsilon_n}(x) \to yf_{P,\lambda}(x)$ since the margin is a linear functional of f. As L is continuous with respect the margin, we obtain $L(y, f_{\varepsilon_n}(x)) \to$

$L(y, f_{P,\lambda}(x))$ for all $(x, y) \in X \times Y$. Recall that $|L(y, f_{\varepsilon_n}(x))|$ is uniformly bounded by the boundedness assumption of $|f|$ and the continuity of L. Therefore, the *bounded convergence theorem* (as a special case of *Lebesgue dominated convergence theorem*) implies

$$\begin{aligned} R_{L,P}(f_{\varepsilon_n}(x)) &= \int_{(x,y)\sim P} L(y, f_{\varepsilon_n}(x)) P_X(dx) P_Y(dy) \\ &\to \int_{(x,y)\sim P} L(y, f_{P,\lambda}(x)) P_X(dx) P_Y(dy) \\ &= R_{L,P}(f_{P,\lambda}(x)). \end{aligned}$$

C. By $R_{L,P}(f_{\varepsilon_n}) \to R_{L,P}(f_{P,\lambda})$, for a fixed $\rho > 0$, there exists an index n_0 such that for all $n \geq n_0$ we have both $\varepsilon_n \leq \rho$ and $R_{L,P}(f_{P,\lambda}) - R_{L,P}(f_{\varepsilon_n}) \leq \rho$. In other words, we obtain the following inequalities

$$\begin{aligned} S(\lambda, \|f_{\varepsilon_n}\|_{EE}) + R_{L,P}(f_{P,\lambda}) - \rho &\leq S(\lambda, \|f_{\varepsilon_n}\|_{EE}) + R_{L,P}(f_{\varepsilon_n}) = R^{reg}_{L,P,\lambda}(f_{\varepsilon_n}) \\ &\leq \inf_{f \in L^1(X)} R^{reg}_{L,P,\lambda}(f) + \varepsilon_n \\ &\leq R^{reg}_{L,P,\lambda}(f_{P,\lambda}) + \varepsilon_n \\ &= S(\lambda, \|f_{P,\lambda}\|_{EE}) + R_{L,P}(f_{P,\lambda}) + \varepsilon_n, \end{aligned}$$

where the second inequality is based on the definition of f_{ε_n}. It implies that

$$S(\lambda, \|f_{\varepsilon_n}\|_{EE}) \leq S(\lambda, \|f_{P,\lambda}\|_{EE}) + \varepsilon_n + \rho \leq S(\lambda, \|f_{P,\lambda}\|_{EE}) + 2\rho.$$

On the other hand, we need to consider another inequality in the opposite direction. By the weak convergence we already have $\|f_{P,\lambda}\|_{BV} \leq \liminf_n \|f_{\varepsilon_n}\|_{BV}$ and $\|f_{P,\lambda}\|_{L^1} \leq \liminf_n \|f_{\varepsilon_n}\|_{L^1}$. However these two inequalities have nothing to do with $\|f\|_{EE}$. Thanks to the lower semicontinuity of the mean curvature's L^p norm, Leonardi and Masnou (2009, Theorem 4.4) proved that

$$\mathcal{F}_p(f) = \int_X |\nabla f|(1 + |\nabla \cdot \left(\frac{\nabla f}{|\nabla f|} \right)|^p) dx$$

is lower semicontinuous in the class of $C^2(\mathbb{R}^d)$ functions whenever $p \geq 1$ for $d = 2$ or $p \geq 2$ for $d \geq 3$. An earlier result (Ambrosio and Masnou, 2003, Theorem 6) required $p > d-1$ for $d \geq 2$. Of course the definition of $\mathcal{F}_p(f)$ is valid only for a certain class of smooth functions and we use the following relaxed functional (Ambrosio and Masnou, 2003; Leonardi and Masnou, 2009)

$$\overline{\mathcal{F}}_p(f) = \inf\{\liminf_{h\to\infty} \mathcal{F}_p(f_h) : f_h \to f \in L^1\}$$

to extend to the whole space $L^1(\mathbb{R}^d)$ (including $BV(X)$). We also have lower semicontinuity of $\overline{\mathcal{F}}_p(f)$ (Ambrosio and Masnou, 2003, Theorem 5) and $\overline{\mathcal{F}}_p(f) = \mathcal{F}_p(f)$ whenever $f \in C^2(X)$ (Leonardi and Masnou, 2009, Theorem 4.4). Immediately we obtain

$$\|f_{P,\lambda}\|_{EE} \leq \liminf_n \|f_{\varepsilon_n}\|_{EE}$$

and thus by the increasing property of $S(\lambda, t)$,

$$S(\lambda, \|f_{P,\lambda}\|_{EE}) \leq \lim_{n\to\infty} S(\lambda, \|f_{\varepsilon_n}\|_{EE}).$$

Combining the inequalities in two directions together yields

$$\lim_{n\to\infty} S(\lambda, \|f_{\varepsilon_n}\|_{EE}) = S(\lambda, \|f_{P,\lambda}\|_{EE}).$$

D. Combining $R_{L,P}(f_{\varepsilon_n}) \to R_{L,P}(f_{P,\lambda})$ with $S(\lambda, \|f_{\varepsilon_n}\|_{EE}) \to S(\lambda, \|f_{P,\lambda}\|_{EE})$, we have

$$R^{reg}_{L,P,\lambda}(f_{\varepsilon_n}) \to R^{reg}_{L,P,\lambda}(f_{P,\lambda}).$$

Because the definition of $\{f_{\varepsilon_n}\}$ indicates

$$R^{reg}_{L,P,\lambda}(f_{\varepsilon_n}) \to \inf_{f\in L^1(X)} R^{reg}_{L,P,\lambda}(f),$$

we have found a $f_{P,\lambda} \in BV(X) \subset L^1(X)$ such that

$$R^{reg}_{L,P,\lambda}(f_{P,\lambda}) = \inf_{f\in L^1(X)} R^{reg}_{L,P,\lambda}(f).$$

E. The second assertion $\|f_{P,\lambda}\|_{EE} \le \delta_\lambda$ is obtained by the boundedness of f_ε in the first step. ■

5.3 Binary Classification Consistency

In classical statistics, a statistic $\hat{\theta}_n$ is a consistent estimator of a parameter θ based on a sample of size n if and only if for any $\varepsilon > 0$, $\lim_{n\to\infty} P(|\hat{\theta}_n - \theta| > \varepsilon) = 0$. In the same spirit, it is natural to request that a learning algorithm should eventually "converge" to an optimal solution when more and more training examples are presented. In the literature of machine learning, there exists two different types of consistency depending on the optimal solution that belongs to some particular function space or the space of all functions (von Luxburg and Schölkopf, 2008). The latter is often called *Bayes consistency* if the risk of a learned classifier converges to the risk of the Bayes optimal decision rule. It is well accepted that a good learning algorithm should satisfy this asymptotic property of consistency when the data size is sufficiently large.

The literature on the consistency analysis of learning algorithms can be roughly classified into following categories: (1) binary classification (Zhang, 2004a; Bartlett et al., March 2006), in particular for SVM (Steinwart, 2005), for Boosting (Bartlett and Traskin, 2007), and for random forests (Biau et al., 2008); (2) multi-class classification (Zhang, 2004b; Tewari and Bartlett, 2007; Glasmachers, 2010); (3) regression (Zakai and Ritov, 2009); (4) learning to rank (Cossock and Zhang, 2008; Xia et al., 2008; Duchi et al., 2010); (5) multi-label learning (Gao and Zhou, 2013). The work by Biau et al. (2008) showed that some popular classifiers, including Breiman's random forest classifier, are not consistent.

We first formalize the definitions of several kinds of consistency used in this section, following von Luxburg and Schölkopf (2008) and Steinwart (2005).

Definition 8 *A classifier f_n is said to be (Bayes)* ***consistent*** *with respect to a given probability measure P if the risk $R(f_n)$ converges in probability to the Bayes risk, that is for all $\varepsilon > 0$,*

$$P(R(f_n) - R(f^*) > \varepsilon) \to 0 \ \ as \ \ n \to \infty$$

where $R(f) \doteq P(\{(x,y) : \text{sign} f(x) \neq y\})$ *is the risk of a classifier* f *and* f^* *denotes the Bayes classifier. Furthermore,* f_n *is said to be* ***universally consistent*** *if it is consistent for all distributions* P *on* $X \times Y$*. It is called* ***strongly universally consistent*** *if such limiting property even holds almost surely (a.s.), that is*

$$P(\lim_{n\to\infty} R(f_n) = R(f^*)) = 1.$$

Note that the Bayes risk is the minimum that we can achieve in the space of all measurable functions, so we always have $R(f_n) \geq R(f^*)$ and there is no need to use the absolute value as in classical statistics.

We also need the notion of simple functions to approximate any function from $L^p(X)$.

Definition 9 *A* ***simple function*** *is a function* $\psi : X \to \mathbb{R}$ *of the form*

$$\psi(\mathbf{x}) = \sum_{i=1}^{n} c_i \chi_{A_i}(\mathbf{x})$$

where χ_A *is the indicator function of the set* A *and* $\{c_i\} \subset \mathbb{R}$*. Another description of a simple function is a function that takes on finitely many values in its range.*

Proposition 10 (From Regularized to Unregularized) *For every Borel probability measure* P *on* $X \times Y$*, we have*

$$\lim_{\lambda\to 0} R^{reg}_{L,P,\lambda}(f_{P,\lambda}) = R_{L,P}$$

where $f_{P,\lambda} \in \text{BV}(X)$ *minimizes the regularized L-risk* $R^{reg}_{L,P,\lambda}(f)$*, and* $R_{L,P}$ *is the smallest possible L-risk* $R_{L,P}(f)$ *achieved by any measurable function* $f : X \to \mathbb{R}$*.*

Proof First by the definition of $f_{P,\lambda}$ we have

$$\begin{aligned}
\lim_{\lambda\to 0} R^{reg}_{L,P,\lambda}(f_{P,\lambda}) &= \lim_{\lambda\to 0} \inf_{f\in \text{BV}(X)} R^{reg}_{L,P,\lambda}(f) \\
&= \lim_{\lambda\to 0} \inf_{f\in \text{BV}(X)} \{S(\lambda, \|f\|_{EE}) + R_{L,P}(f)\} \\
&= \inf_{f\in \text{BV}(X)} \{\lim_{\lambda\to 0} S(\lambda, \|f\|_{EE}) + R_{L,P}(f)\} \\
&= \inf_{f\in \text{BV}(X)} R_{L,P}(f)
\end{aligned}$$

since $S(\lambda, \cdot)$ is continuous in 0 with respect to λ and $S(0, \cdot) = 0$. Next we show that the following identities hold true

$$\inf_{f\in \text{BV}(X)} R_{L,P}(f) = \inf_{f\in L^1(X)} R_{L,P}(f) = R_{L,P}$$

for a sequence of embedding spaces $\text{BV}(X) \subset L^1(X) \subset \{f : X \to \overline{\mathbb{R}} \text{ measurable}\}$, which suffices to prove the assertion.

We first check the first identity. Recall that the simple functions that belong to $L^p(X)$ are *dense* in $L^p(X)$ for $1 \leq p \leq \infty$ (Hunter, 2011, Theorem 7.8). Note that an integrable simple function

$$\psi = \sum_{i=1}^{n} c_i \chi_{A_i}$$

belongs to $L^p(X)$ for $1 \leq p < \infty$ if and only if $\mu(A_i) < \infty$ for each $A_i \subset X$ such that $c_i \neq 0$, meaning that its support has finite μ measure. On the other hand, each simple function belongs to L^∞. We restrict the discussion on bounded functions in $L^p(X)$ since any unbounded $f \in L^p(X)$ can be replaced by a modified bounded $\tilde{f} \in L^p(X)$ to make the loss L smaller. Hence the nice property of density indicates that for every bounded $f \in L^p(X)$ $(1 \leq p \leq \infty)$, there exists a sequence of simple functions g_n such that $\|f - g_n\|_{L^p} \to 0$ and $|g_n(x)| \leq |f(x)|$ pointwise. The strong convergence in L^p norm implies the weak convergence in measure

$$P_X(\{x \in X : |f - g_n| \geq \varepsilon\}) \to 0.$$

Since $L(y,t)$ is uniformly continuous with respect to the second variable in the closed interval $[-|f(x)|, |f(x)|]$, for any fixed y we have

$$P_X(\{x \in X : |L(y, f(x)) - L(y, g_n(x))| \geq \varepsilon\}) \to 0.$$

By the previous assumption that $L(y, f(x))$ is a margin-based admissible loss function which is continuous with respect to the margin $yf(x)$, there exists a function $\hat{L}(yf(x)) \in L^1(X)$ such that

$$|L(y, g_n(x))| \leq \hat{L}(yf(x)).$$

By the *Lebesgue's dominated convergence theorem*, the expectation in $R_{L,P}(f)$ and the limit can change order:

$$\begin{aligned} \lim_{n\to\infty} \int_{(x,y)\sim P} L(y, g_n(x)) P_X(dx) P_Y(dy) &= \int_{(x,y)\sim P} L(y, f(x)) P_X(dx) P_Y(dy) \\ &= \mathbb{E}_{(x,y)\sim P} L(y, f(x)) \\ &= R_{L,P}(f). \end{aligned}$$

Thus by fixing $p = 1$ we have

$$\inf\{R_{L,P}(f) : f \text{ simple}\} = \inf_{f \in L^1(X)} R_{L,P}(f).$$

Clearly such simple functions belong to $\mathrm{BV}(X)$, and also by the definition of BV functions we have $\mathrm{BV}(X) \subset L^1(X)$. Then the relation of embedding spaces implies that

$$\inf\{R_{L,P}(f) : f \text{ simple}\} \geq \inf_{f \in \mathrm{BV}(X)} R_{L,P}(f) \geq \inf_{f \in L^1(X)} R_{L,P}(f).$$

Together with the previous identity between simple functions and $L^1(X)$ functions, the first identity

$$\inf_{f \in \mathrm{BV}(X)} R_{L,P}(f) = \inf_{f \in L^1(X)} R_{L,P}(f)$$

follows.

The second identity comes from the fact

$$\inf_{f \in L^\infty(X)} R_{L,P}(f) = R_{L,P}$$

with the proof given by Steinwart (2005, Proposition 3.2). On the other hand, the embedding relationship $L^\infty(X) \subset L^1(X) \subset \{f : X \to \overline{\mathbb{R}} \text{ measurable}\}$ leads to

$$\inf_{f \in L^\infty(X)} R_{L,P}(f) \geq \inf_{f \in L^1} R_{L,P}(f) \geq R_{L,P}.$$

Therefore the second identity

$$\inf_{f \in L^1} R_{L,P}(f) = R_{L,P}$$

holds true. ■

Following the framework of consistency proof in Steinwart (2005), we need the final piece of the puzzle by showing that some suitable concentration inequalities hold true for our proposed algorithms. These concentration inequalities bridge the gap between the expected L-risk of $f_{P,\lambda}$ and the empirical L-risk of $f_{P,\lambda}$. Steinwart's framework is somehow modular: each tuple of concentration inequality, loss function, and function space gives a condition on $\{\lambda_n\}$ ensuring $|R_{L,P}(f_{P,\lambda}) - R_{L,T}(f_{P,\lambda})| \to 0$, and each different combination of this tuple leads to new consistency results. There exist several concentration inequalities in Steinwart (2005) based on covering numbers, localized covering numbers, and algorithmic stability. Among these three concentration inequalities, the algorithmic stability (Bousquet and Elisseeff, 2002; Kutin and Niyogi, 2002; Poggio et al., 2004) is an elegant approach that does not depend on any complexity measure of the underlying hypothesis space, but rather depend on how the learning algorithm searches this space. However, stability based concentration inequalities (Bousquet and Elisseeff, 2002) heavily rely on the reproducing property of the RKHS space and often require that the regularization term is convex, while these conditions do not hold for our elastica based learning algorithm. In the following we give a concentration inequality based on covering numbers.

For a metric space (M, d) we define its *covering number* $\mathcal{N}((M, d), \varepsilon)$ to be the minimal l such that there exist l disks in M with radius ε covering M:

$$\mathcal{N}((M,d),\varepsilon) \doteq \min\Big\{\, l \in \mathbb{N} \,:\, \{x_1,\ldots,x_l\} \subset M,\ M \subset \bigcup_{i=1}^{l} B(x_i,\varepsilon)\Big\},$$

where $B(x, \varepsilon)$ denotes the closed ball with center x and radius $\varepsilon \geq 0$. We also have to measure the continuity of a given loss function L. The *modulus of continuity* of L is defined by

$$\omega(L,\delta) \doteq \sup\{|L(y,t) - L(y,t')| \,:\, y \in Y,\ t,t' \in \mathbb{R},\ |t-t'| \leq \delta\}.$$

In addition we define the *inverted modulus of continuity* as

$$\omega^{-1}(L,\varepsilon) \doteq \sup\{\delta > 0 :\ \omega(L,\delta) \leq \varepsilon\}.$$

Moreover, since only $f_{P,\lambda} \in \mathrm{BV}(X)$ and $f_{T,\lambda} \in \mathrm{BV}(X)$ are our focus considered in the consistency results, we define the *restricted loss function*:

$$L_\lambda(\cdot,\cdot) \doteq L(y,f(x)) :\ y \in Y,\ f \in \mathrm{BV}(X) \cap L^\infty(X),\ \|f\|_{TV} \leq \delta_\lambda\,,$$

where δ_λ given in Theorem 7 is a simple upper bound on the TV semi-norm of the solutions of $R^{reg}_{L,P,\lambda}(f)$.

Lemma 11 (Concentration) *For all Borel probability measures P on $X \times Y$, all $\varepsilon > 0$, $\lambda > 0$, and all $n \geq 1$ we have*

$$P(|R_{L,T}(f_{T,\lambda}) - R_{L,P}(f_{T,\lambda})| \geq \varepsilon) \;\leq\; 2\mathcal{N}\left(\delta_\lambda I,\, \omega^{-1}(L_\lambda, \varepsilon/3)\right) \exp\Big(-\frac{2n\varepsilon^2}{9\|L_\lambda\|_\infty^2}\Big),$$

where $\delta_\lambda I \doteq \{f \in \mathrm{BV}(X) \cap L^\infty(X) : \|f\|_{TV} \leq \delta_\lambda\}$ is a metric space equipped with the $\|\cdot\|_\infty$ norm.

Proof Write the loss class as $\mathcal{F} \doteq \{L(\cdot, f(\cdot)) : f \in \mathrm{BV}(X) \cap L^\infty(X),\ \|f\|_{TV} \leq \delta_\lambda\}$. Note that $\mathcal{F}$ is a subset of $C(X \times Y)$ of nonnegative functions that are bounded by $\|L_\lambda\|_\infty$. Let $l = \mathcal{N}(\mathcal{F}, \varepsilon/3)$ and consider $f_1, \ldots, f_l$ such that the disks D_j centered at f_j and with radius $\varepsilon/3$ cover F. Recall that *Hoeffding's inequality* (Bousquet et al., 2004, Theorem 1) (see also the book by Boucheron et al., 2013), perhaps the most elegant quantitative version of the law of large numbers, states that for all $\varepsilon > 0$,

$$P\Big(\Big|\frac{1}{n}\sum_{i=1}^{n} f(Z_i) - \mathbb{E}[f(Z)]\Big| > \varepsilon\Big) \leq 2\exp\Big(-\frac{2n\varepsilon^2}{(b-a)^2}\Big),$$

where $Z_1, \ldots, Z_n$ be n i.i.d. random variables with $f(Z) \in [a, b]$. For each fixed f_j, applying Hoeffding's inequality yields

$$P(|R_{L,T}(f_j) - R_{L,P}(f_j)| \leq \varepsilon/3) \geq 1 - 2\exp\Big(-\frac{2n(\varepsilon/3)^2}{\|L_\lambda\|_\infty^2}\Big),$$

with $R_{L,P}(f_j) = \mathbb{E}_{(x,y)\sim P} L(y, f_j(x))$ and $L(y, f_j(x)) \in [0, \|L_\lambda\|_\infty]$. As the disks D_j are $\varepsilon/3$ cover of F, the following inequalities hold true

$$\begin{aligned}
& \sup_{f \in Dj} |R_{L,T}(f) - R_{L,P}(f)| \\
= \;& \sup_{f \in Dj} |R_{L,T}(f) - R_{L,T}(f_j) + R_{L,T}(f_j) - R_{L,P}(f_j) + R_{L,P}(f_j) - R_{L,P}(f)| \\
\leq \;& \varepsilon/3 + |R_{L,T}(f_j) - R_{L,P}(f_j)| + \varepsilon/3 \\
\leq \;& \varepsilon\,,
\end{aligned}$$

with probability at least $1 - 2\exp(-\frac{2n\varepsilon^2}{9\|L_\lambda\|_\infty^2})$ over the random choice of the training set T. Since $\|f\|_{EE} \leq \delta_\lambda$ implies $\|f\|_{TV} \leq \delta_\lambda$, using the union bound we get

$$P(\sup_{\|f\|_{EE} \leq \delta_\lambda} |R_{L,T}(f) - R_{L,P}(f)| \geq \varepsilon) \leq 2\mathcal{N}(\mathcal{F}, \varepsilon/3)\exp(-\frac{2n\varepsilon^2}{9\|L_\lambda\|_\infty^2}).$$

By the definition of the modulus of continuity, every ε cover $f_1, \ldots, f_l$ with $\|f_j\|_{TV} \leq \delta_\lambda$ defines an $\omega(L_\lambda, \varepsilon)$ cover $L(\cdot, f_1(\cdot)), \ldots, L(\cdot, f_l(\cdot))$ of $\mathcal{F}$ with respect to the supremum norm. Thus we have

$$\mathcal{N}(\mathcal{F}, \varepsilon/3) \leq \mathcal{N}(\delta_\lambda I,\, \omega^{-1}(L_\lambda, \varepsilon/3)),$$

which immediately yields

$$P\Big(\sup_{f \in \delta_\lambda I} |R_{L,T}(f) - R_{L,P}(f)| \geq \varepsilon\Big) \leq 2\mathcal{N}(\delta_\lambda I, \omega^{-1}(L_\lambda, \varepsilon/3))\exp\Big(-\frac{2n\varepsilon^2}{9\|L_\lambda\|_\infty^2}\Big).$$

Since Lemma 7 guarantees that $\|f_{P,\lambda}\|_{TV} \leq \delta_\lambda$ or $\|f_{T,\lambda}\|_{TV} \leq \delta_\lambda$, the assertion follows. ■

Theorem 12 (Universal Consistency) *The classifier $f_{T,\lambda_n} \in \mathrm{BV}(X)$ minimizing the regularized empirical L-risk $R^{reg}_{L,T,\lambda_n}(f)$ is universally consistent for a positive sequence $\{\lambda_n\}$ with $\lambda_n \to 0$ and*

$$\frac{1}{n}\|L_{\lambda_n}\|^2_\infty \ln \mathcal{N}(\delta_{\lambda_n} I, \omega^{-1}(L_{\lambda_n}, \varepsilon)) \to 0$$

for all $\varepsilon > 0$.

Proof The Proposition 3.3 of Steinwart (2005) states that for any Borel probability measure P on $X \times Y$ and for all $\varepsilon > 0$, there exists a $\delta > 0$ such that for all measurable $f : X \to \overline{\mathbb{R}}$ with $R_{L,P}(f) \leq R_{L,P} + \delta$ we have $R_P(f) \leq R_P + \varepsilon$. Here L in $R_{L,P}(f)$ requires to be an admissible loss function. Therefore, in order to prove the 0-1 risk $R_P(f_{T,\lambda_n}) \leq R_P + \varepsilon$, it suffices to show the L-risk $R_{L,P}(f_{T,\lambda_n}) \leq R_{L,P} + \delta$.

The outline is given as follows:

$$\begin{aligned}
R_{L,P}(f_{T,\lambda_n}) &\leq S(\lambda_n, \|f_{T,\lambda_n}\|_{EE}) + R_{L,P}(f_{T,\lambda_n}) \\
&\leq S(\lambda_n, \|f_{T,\lambda_n}\|_{EE}) + R_{L,T}(f_{T,\lambda_n}) + \delta/3 \qquad (27)\\
&\leq S(\lambda_n, \|f_{P,\lambda_n}\|_{EE}) + R_{L,T}(f_{P,\lambda_n}) + \delta/3 \qquad (28)\\
&\leq S(\lambda_n, \|f_{P,\lambda_n}\|_{EE}) + R_{L,P}(f_{P,\lambda_n}) + 2\delta/3 \qquad (29)\\
&= R^{reg}_{L,P,\lambda_n}(f_{P,\lambda_n}) + 2\delta/3 \\
&\leq R_{L,P} + \delta. \qquad (30)
\end{aligned}$$

Among the above inequalities, (27) and (29) hold true by the empirical concentration inequality in Lemma 11 with probability at least

$$1 - 2\mathcal{N}\left(\delta_\lambda I,\, \omega^{-1}(L_\lambda, \varepsilon/3)\right) \exp\Big(-\frac{2n\varepsilon^2}{9\|L_\lambda\|^2_\infty}\Big)$$

over the random choice of the training set T, while (28) is obtained by the fact that f_{T,λ_n} minimizes the regularized empirical L-risk $R^{reg}_{L,T,\lambda_n}(f)$. Proposition 10 with respect to $\lambda_n \to 0$ immediately implies (30): there exists an integer $n_0 \geq 1$ such that for all $n \geq n_0$ we have

$$|R^{reg}_{L,P,\lambda_n}(f_{P,\lambda_n}) - R_{L,P}| \leq \delta/3.$$

Note that the condition

$$\frac{1}{n}\|L_{\lambda_n}\|^2_\infty \ln \mathcal{N}(\delta_{\lambda_n} I, \omega^{-1}(L_{\lambda_n}, \varepsilon)) \to 0$$

assures that $R_{L,P}(f_{T,\lambda_n}) \leq R_{L,P} + \delta$ holds true with probability 1 nearly as $n \to \infty$. Then the universal consistency follows by $P(R_P(f_{T,\lambda_n}) - R_P \leq \varepsilon) \to 1$ for all distributions P on $X \times Y$. ■

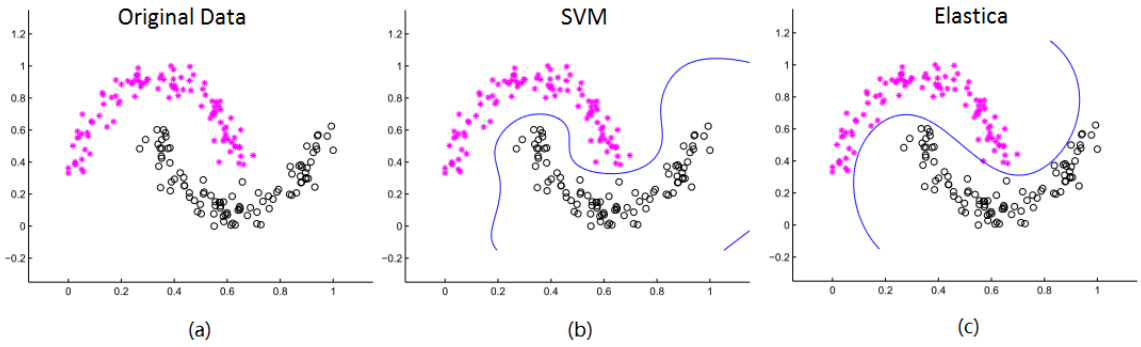


Figure 2: Decision boundaries produced by SVM and EE with common parameters on two moon data.

6. Experimental Results

The proposed two models (TV and EE) are compared with LR, SVM with RBF kernels using the LIBSVM implementation (Chang and Lin, 2011), and Back-Propagation Neural Networks (BPNN) in the Matlab neural network toolbox. Two implementations of our methods are also compared: Gradient Descent method (GD) and Lagged Linear Equation method (LAG). The maximum number of iterations in GD and LAG is empirically setting as 40. Binary classification, multi-class classification, and regression tasks are tested on synthetic and real-world data sets. We collected real data sets from the libsvm website (Chang and Lin, 2011) and the UCI machine learning repository (Asuncion and Newman, 2013). Some attributes have been removed due to missing entries. Some data sets have a huge number of instances, hence we use only 1000 instances in our experiments. All data sets are scaled into [0,1] before training and testing.

6.1 Synthetic Data

We first compare our EE model and SVM for binary classification on two synthetic data sets: the two moon data and one data set made by ourselves. Fig. 2 and Fig. 3 show the decision boundaries produced by SVM and EE with common parameters. We can see that SVM tends to yield curved or even wiggly decision boundaries to pursue low training errors. In contrast, smooth or even straight decision boundaries with low curvature are favored by EE, hence reducing the risk of overfitting.

One may argue that SVM can produce smooth and low curvature decision boundaries by tuning the parameters. Fig. 4 shows the results of SVM with different combinations of kernel parameter g and slack parameter C. For comparison, Fig. 5 displays the results of EE with different combinations of regularization parameter λ and kernel parameter c. We can see that most decision boundaries produced by EE have lower curvature values and are smoother than the results by SVM. Actually the elastica term in EE may be interpreted as the accumulated bending energy of all level lines, including the level line on the decision boundary.

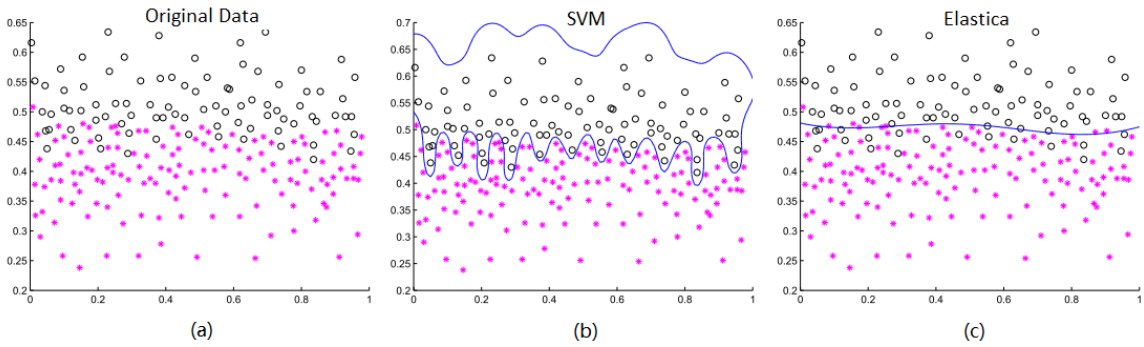


Figure 3: Decision boundaries produced by SVM and EE with common parameters on our synthetic data.

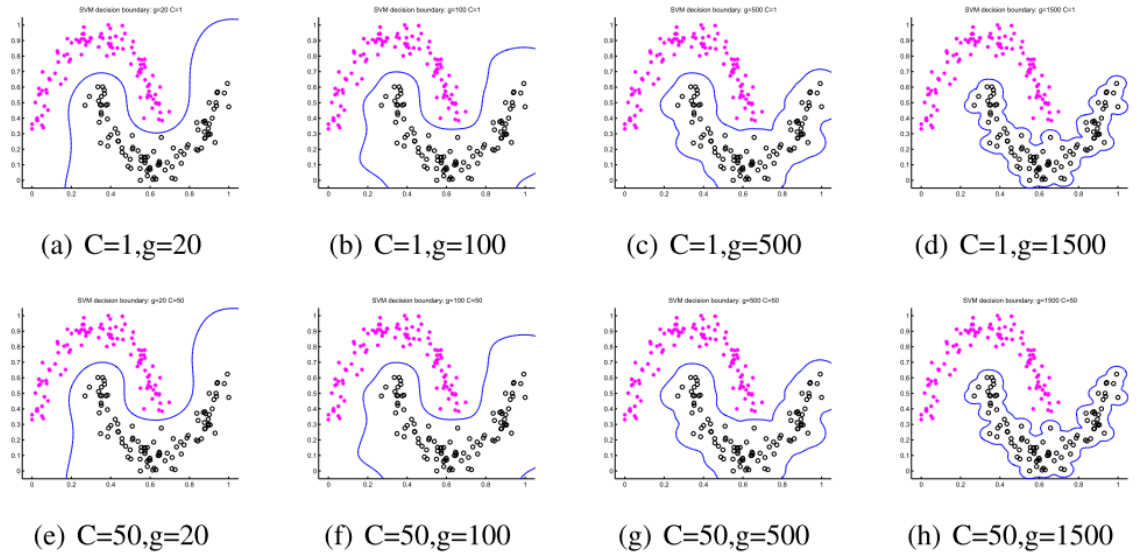
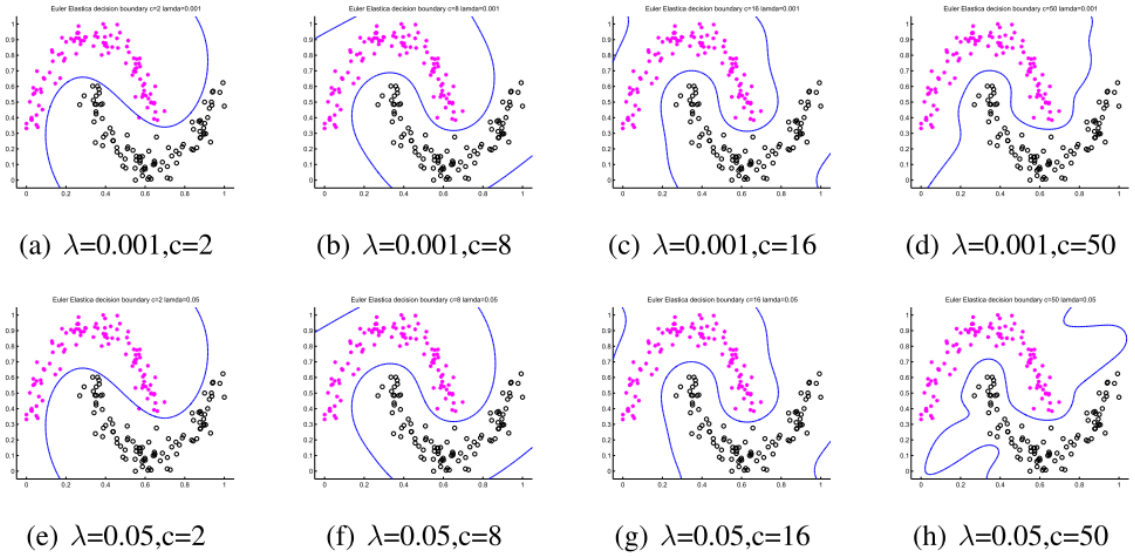


Figure 4: Decision boundaries produced by SVM with different parameter combinations on two moon data.

6.2 Binary Classification

We use eleven data sets for binary classification. The optimal parameters for each algorithm are selected by grid search using 5-fold cross-validation. To make the grid search more practical, only two common parameters are searched for all methods except BPNN: (C, g) for SVM, while (c, λ) for LR, TV, and EE. Empirically, the parameter η is set as 1 for LR, and the parameter b is fixed as 0.01 for EE. Then excluding BPNN, the two common parameters are searched from $-10 : 10$ in logarithm with step 2. For each data set, we randomly run the 5-fold cross validation ten times to reduce the influence of data partitions.



(a) λ=0.001,c=2 (b) λ=0.001,c=8 (c) λ=0.001,c=16 (d) λ=0.001,c=50

(e) λ=0.05,c=2 (f) λ=0.05,c=8 (g) λ=0.05,c=16 (h) λ=0.05,c=50

Figure 5: Decision boundaries produced by EE with different parameter combinations on two moon data.

Table 1 gives the average classification accuracies (with standard deviations) for the five methods. The results indicate that BPNN performs the worst, while the LAG version of EE achieves the best accuracies on six data sets. LR and other implementations of TV and EE are comparable with SVM. When comparing EE-LAG and SVM in a pairwise fashion, we can see that EE-LAG achieves improvements over SVM on 10 datasets (though not much statistically significant as the differences on two averaged accuracies is often less than one standard deviation).

6.3 Multi-Class Classification

For multi-class tasks, we collected twelve data sets. For the 256-dimensional USPS data, PCA is used as a preprocessing step to reduce the dimension to 30 and we randomly select 1000 samples for experiments. Same as the settings for binary problems, we use ten runs 5-fold cross-validation to choose the optimal parameters for each method. All methods except for BPNN have two common parameters which are searched from $-10 : 10$ in logarithm with step 1.

Aside from BPNN that has a built-in ability for multi-class tasks, almost all function learning approaches are originally designed for binary classification. In order to handle multi-class situations, usually "one versus all" (OVA) or "one versus one" (OVO) strategies can be adopted. If using OVA, one needs to learn M scoring functions to fulfill the multi-class task, where M is the number of classes. The final decision is the label whose scoring function achieves the largest value or confidence score. However, these scoring functions are learned independently, often suffering to the so-called *calibration problem* (Mohri et al., 2012, chap. 8). LIBSVM uses the OVO strategy, with some reasons and detailed comparisons given in (Hsu and Lin, 2002). See also Mohri et al. (2012, chap. 8) for dis-

Data	Dim	Num	SVM	BPNN	LR	TV		EE	
						GD	LAG	GD	LAG
Australian	14	690	85.94 ±2.70	85.34 ±1.97	87.06 ±2.45	87.11 ±2.06	87.01 ±2.46	86.54 ±2.31	**87.25** ±2.01
Blood transfusion	4	748	79.01 ±3.01	79.08 ±3.36	79.32 ±3.74	79.55 ±2.38	79.42 ±2.60	**79.73** ±2.18	**79.73** ±2.03
Breast-cancer	10	683	97.36 ±1.59	96.40 ±1.14	97.60 ±1.27	97.36 ±1.28	97.72 ±1.43	97.13 ±1.37	**97.83** ±1.29
Diabetes	8	768	77.73 ±3.03	76.85 ±4.22	77.96 ±3.50	77.83 ±3.19	77.81 ±2.73	**78.23** ±2.54	78.10 ±2.63
German. number	24	1000	77.10 ±1.61	76.37 ±1.61	77.10 ±1.36	76.19 ±1.47	77.10 ±1.29	76.50 ±1.59	**77.22** ±1.30
Haberman's survival	3	306	74.51 ±4.31	74.52 ±3.53	**75.77** ±3.00	75.30 ±3.31	75.28 ±3.78	75.65 ±3.42	75.34 ±3.32
Heart	13	270	83.70 ±2.72	81.76 ±3.16	84.26 ±2.22	84.45 ±2.82	84.58 ±2.73	84.78 ±2.69	**84.96** ±2.79
Liver-disorders	6	345	73.62 ±5.72	71.52 ±4.44	73.20 ±2.95	**74.81** ±2.49	73.62 ±2.65	74.32 ±2.29	73.91 ±2.83
Planning relax	12	182	**73.63** ±4.41	67.62 ±4.93	72.22 ±4.46	71.67 ±4.93	71.67 ±4.08	72.22 ±4.25	71.67 ±4.79
Sonar	60	208	89.90 ±4.41	88.99 ±4.79	**90.88** ±3.83	90.30 ±4.47	90.27 ±4.72	90.07 ±3.27	90.50 ±3.37
Vertebral column	6	310	85.81 ±4.26	85.16 ±3.12	84.52 ±3.90	84.55 ±4.14	84.75 ±4.37	85.83 ±3.38	**85.92** ±3.68

Table 1: Average accuracies (%) for binary classification with 5-fold cross-validation.

cussions between OVA and OVO. Recently in Varshney and Willsky (2010), an efficient binary encoding strategy was proposed to represent the decision boundary by using only $m = \lceil log_2 M \rceil$ functions. Empirically we compared the $log_2 M$ strategy and the OVA strategy for LR, TV and EE, and found that the in most cases the $log_2 M$ strategy performs slightly better. As the codewords for making decisions are represented as 0-1 bits of length m, the $log_2 M$ strategy may somehow "favor" those methods with good function approximation ability. In multi-class experiments, the $log_2 M$ strategy is used for LR, TV and EE, while LIBSVM runs with the OVO strategy.

The multi-class results of classification accuracies are shown in Table 2. The accuracy results demonstrate that both SVM and EE-GD offer the best accuracies on four (different) data sets, and both EE-LAG and TV-GD take the first place on two (different) data sets. If we compare SVM and EE-GD in a pairwise fashion by excluding other competing methods, the results show that SVM wins on only five data sets while EE-GD performs better on the other seven data sets. Therefore on multi-class tasks, Table 2 implies that our EE-GD version can offer competitive results, or can perform slightly better than SVM.

6.4 Regression

We use ten regression data sets to validate the proposed TV/EE methods compared with SVM, BPNN, and LR. All data sets are scaled into [0,1]. The same experimental settings

Data	Cls	Dim	Num	SVM	BPNN	LR	TV		EE	
							GD	LAG	GD	LAG
Balance scale	3	4	625	**98.40** ±1.13	92.48 ±1.77	89.44 ±1.90	90.88 ±1.25	89.92 ±1.36	90.40 ±1.80	91.36 ±1.35
Flags	8	29	194	52.06 ±7.57	46.90 ±7.25	53.13 ±7.34	51.50 ±7.29	52.10 ±7.10	**53.55** ±6.39	52.10 ±7.22
Glass	6	9	214	73.83 ±9.22	63.99 ±11.83	73.81 ±7.34	75.59 ±8.73	**76.19** ±8.62	75.82 ±9.07	75.71 ±9.15
Hayes-rath	3	5	132	**81.82** ±4.12	74.26 ±4.62	73.63 ±4.31	77.87 ±4.59	77.08 ±4.67	78.90 ±4.29	78.15 ±4.31
Iris	3	4	150	**96.67** ±3.65	96.00 ±3.37	95.33 ±3.42	96.00 ±3.50	96.00 ±3.27	96.00 ±3.33	96.00 ±3.10
Statlog imageseg	7	19	2310	97.27 ±0.91	96.74 ±1.12	97.31 ±0.95	97.31 ±0.93	97.21 ±0.88	**97.45** ±0.81	97.44 ±0.83
Seeds	3	7	210	94.76 ±1.78	**95.71** ±1.56	92.38 ±1.85	92.86 ±1.74	92.65 ±1.62	92.86 ±1.93	92.75 ±1.87
Teaching assist	3	5	151	60.93 ±20.97	56.63 ±19.44	63.47 ±17.28	65.18 ±14.26	66.00 ±15.37	65.41 ±16.23	**67.33** ±17.41
USPS	10	30	1000	94.10 ±1.39	89.72 ±2.79	94.90 ±1.28	94.40 ±1.32	94.80 ±1.73	94.40 ±1.54	**95.00** ±1.27
Vehicle	4	18	846	84.40 ±0.70	79.18 ±1.41	82.75 ±1.33	**85.00** ±0.82	84.25 ±0.93	**85.00** ±0.78	84.84 ±0.90
Wine	3	13	178	98.88 ±1.27	97.78 ±1.43	**99.44** ±0.83	**99.44** ±0.83	99.43 ±0.85	**99.44** ±0.83	98.86 ±1.31
Yeast	10	8	1484	**60.78** ±3.26	54.49 ±4.57	58.22 ±3.79	57.95 ±3.34	57.91 ±3.27	57.95 ±3.64	57.97 ±3.52

Table 2: Average accuracies (%) for multi-class classification with 5-fold cross-validation.

are repeated by running ten times of 5-fold cross-validation for each data set. Table 3 shows the regression results in mean square errors (MSE) with standard deviations.

Clearly, we can see that TV-LAG and two versions of EE achieve the best regression results, with each winning three times on overall ten data sets. BPNN yields the lowest errors on two data sets. Surprisingly SVM takes the first place on only one data set. If we select SVM and LR in a pairwise fashion by excluding other methods, we find that LR offers lower errors on seven data sets while SVM performs better on only other three data sets. If we compare SVM and TV-GD separately by neglecting other methods, TV-GD performs better on nine data sets. Note that TV-GD performs the worst among all versions of TV/EE. These results demonstrate that compared with other competing methods, the performance of SVM on regression tasks is rather unsatisfactory. The reason might be that the original purpose of SVM is designed for classification, not for regression. In contrast, our TV/EE methods exhibit excellent regression ability on these data sets.

6.5 Running Times

To compare the real performance in computational burdens, in Table 4 we list the running times of the competing methods on five data sets for binary classification. The running times are obtained for five-fold cross-validation in one single round, averaged by ten rounds. The experiments are conducted on a PC Sever with two Intel Xeon 5620 cores and 8GB RAM.

Data	Dim	Num	SVM	BPNN	LR	TV		EE	
						GD	LAG	GD	LAG
Auto MPG	7	392	7.11 ±0.56	5.63 ±0.57	6.07 ±0.55	5.67 ±0.56	5.62 ±0.53	**5.47** ±0.51	5.69 ±0.56
Concrete comp. str.	8	1030	6.42 ±0.62	**4.88** ±0.56	6.02 ±0.64	5.98 ±0.83	5.43 ±0.60	5.83 ±0.78	5.24 ±0.61
Concrete slump test	9	103	4.48 ±2.00	14.30 ±7.14	5.01 ±1.70	1.86 ±0.81	**1.61** ±0.70	1.76 ±0.71	**1.61** ±0.70
Forest fires	12	517	5.95 ±3.62	6.20 ±3.73	3.41 ±3.69	3.43 ±3.61	3.41 ±3.69	**3.37** ±3.54	3.41 ±3.69
Housing	13	506	5.88 ±2.28	7.54 ±2.41	5.13 ±2.36	4.92 ±2.47	**4.90** ±2.29	5.14 ±2.39	4.95 ±2.33
Machine CPU	6	209	3.32 ±2.81	5.18 ±3.23	1.78 ±1.70	2.37 ±1.80	1.91 ±1.78	**1.75** ±1.72	**1.75** ±1.72
Pyrim	27	74	9.32 ±9.75	23.06 ±9.97	6.59 ±6.22	**5.81** ±5.17	5.89 ±5.85	5.90 ±6.09	5.93 ±6.12
Servo	4	167	9.93 ±5.09	**5.62** ±5.24	7.29 ±5.80	8.81 ±5.49	8.34 ±5.63	8.87 ±5.29	7.86 ±5.83
Triazines	60	186	**19.24** ±6.61	41.90 ±8.71	20.73 ±4.46	19.67 ±3.93	20.32 ±4.08	19.63 ±2.50	19.95 ±2.79
Yacht hydrodynamics	6	308	4.52 ±0.31	3.70 ±0.29	7.75 ±1.91	2.33 ±0.47	**1.45** ±0.32	2.07 ±0.43	**1.45** ±0.32

Table 3: Regression errors measured by MSE (10^{-3}) with 5-fold cross-validation.

We can see that the computational burdens of TV/EE algorithms is similar to that of BPNN in Matlab toolbox, but much slower than LIBSVM. The computational PDE approach of our TV/EE models is implemented by gradient descent or lagged iteration, which often requires a long time for assuring that the iterations converge. In each iteration, all the data points participate in the computations of our methods within the current implementations. In contrast, the solutions of SVM is essentially sparse, and recent several improvements show that carefully selecting a small representative subset of the training data can further greatly speed-up the optimization process of SVM (Nandan et al., 2014; Wang et al., 2014).

Our intention in this paper is not to develop a fully-fledged and highly optimized algorithm for supervised learning problems. Instead, this work only serves as a starting point for applying Euler's elastica to classification and regression tasks. The above experiments have demonstrated the excellent accuracies of our elastica based algorithms, though the numerical solutions are rather slow. Hence there exists an opportunity to dramatically improve the computational efficiency by considering the following techniques: (1) Some first order numerical methods, like the augmented Lagrangian method (ALM). The operator splitting method and ALM have been successfully implemented to solve Euler's elastica model for image applications (Tai et al., 2011; Hahn et al., 2011; Duan et al., 2013). The speed-up is spectacular compared with prior approaches. Interestingly the ALM has been also applied to optimize the primal SVM problem with linear computational cost (Nie et al., 2014). (2) Imposing the sparsity constraint on the coefficients $\mathbf{w}$. The sparsity property may enhance the efficiency in each iteration. (3) Selecting a small representative subset of the training data in a similar way proposed by Nandan et al. (2014) and Wang et al. (2014).

Data	Dim	Num	SVM	BPNN	LR	TV		EE	
						GD	LAG	GD	LAG
Australian	14	690	0.859	30.673	4.734	24.734	32.453	25.734	33.197
Blood transfusion	4	748	0.297	22.247	5.938	28.467	35.746	27.481	36.497
Breast-cancer	10	683	0.453	20.318	4.609	18.953	19.447	19.732	20.278
Diabetes	8	768	0.547	23.142	6.453	20.120	20.981	22.145	21.519
German.number	24	1000	1.266	31.452	14.156	29.266	32.145	34.497	31.876

Table 4: Running times (in seconds) for binary classification with 5-fold cross-validation in one single round.

7. Conclusion

Regularization framework and function learning approaches have become very popular in the recent machine learning literature. Due to the great success of total variation and Euler's elastica models in image processing area, we extend these two models for supervised classification and regression on high dimensional data sets. The TV regularizer permits steeper edges near the decision boundaries, while the elastica smoothing term penalizes non-smooth level set hypersurfaces of the target function. Compared with SVM and BPNN, our proposed methods have demonstrated the competitive performance on commonly used benchmark data sets. Specifically, TV and EE offer superb results on binary classification and regression tasks, and performs slightly better than SVM on multiclass problems. In comparison, SVM often yields excellent accuracies for multi-class classification, but offer poor results on regression problems.

Our future work is to explore other possibilities in using different basis functions and to speedup the training time. Recently, several fast Augmented Lagrangian Methods (ALM) (Tai et al., 2011; Duan et al., 2013) have been applied to solve Euler's elastica models in image denoising, inpainting, and zooming applications. Particularly in Duan et al. (2013), the Euler's elastica functional is reformulated as a serial of subproblems, which can be efficiently solved by either closed-form solution or fast iteration method. Whether these methods can be extended to high dimensional problems needs further investigations. Another interesting direction is to extend the work of Zakai and Ritov (2009) on regression consistency to the TV and EE models.

Acknowledgments

The authors thank the anonymous reviewers for valuable comments and insightful suggestions. This work was supported by the National Basic Research Program of China (973 Program) 2011CB302202, the Natural Science Foundation of China (NSFC Grants 61075119 and 61375051), and the Seeding Grant for Medicine and Information Sciences of Peking University (2014-MI-21).

Appendix A: Curvature

The following material comes from Aubert and Kornprobst (2006, chap. 2.4) with slightly different notations. Readers are also referred to the classical geometry book of do Carmo (1976).

Let $\mathbf{c}(p) = (x(p), y(p))$ be a regular planar oriented curve on R^2 with parameter $p \in [0, 1]$. Then $\mathbf{T}(p) = \mathbf{c}'(p) = (x'(p), y'(p))$ is the tangent vector, $\mathbf{N}(p) = (-y'(p), x'(p))$ is the normal vector, and

$$s(p) = \int_0^p |\mathbf{c}'(q)|dq = \int_0^p \sqrt{(x'(p))^2 + (y'(p))^2}dq$$

is the arc length. Due to the regularity condition $\mathbf{c}'(p) \neq 0$, the arc length s is a differentiable function of p and $ds/dp = |\mathbf{c}'(p)|$. If we parametrize the regular curve $\mathbf{c}$ by s, then $\mathbf{T}(s) = d\mathbf{c}(s)/ds$ is the unit tangent vector satisfying $|\mathbf{T}(s)| = 1$. The number $\kappa(s) \doteq |d\mathbf{T}(s)/ds|$ is called the *curvature* at s, measuring the change rate of the angle which neighboring tangents make. Since $|\mathbf{T}(s)| = 1$, we have $\mathbf{T}(s) \cdot d\mathbf{T}(s)/ds = 0$, indicating $d\mathbf{T}(s)/ds$ is collinear to the unit normal vector $\mathbf{N}(s)$. That is, under the arc length parametrization, $d\mathbf{T}(s)/ds = \kappa(s)\mathbf{N}(s)$, or $\kappa(s) = |\mathbf{T} \times d\mathbf{T}/ds| = |\mathbf{c}'(s) \times \mathbf{c}''(s)|$ where $\times$ is the exterior product. Back to the general parametrization $\mathbf{c}(p)$, we have

$$\kappa(p) = \frac{|\mathbf{c}'(p) \times \mathbf{c}''(p)|}{|\mathbf{c}'(p)|^3} = \frac{x'y'' - x''y'}{((x')^2 + (y')^2)^{3/2}}. \tag{31}$$

Now we derive the divergence expression (6) of the curvature on a level curve. Consider the case where $\mathbf{c}(s)$ is the l-level curve of a function $u : R^2 \to R$, denoted by

$$\mathbf{c}(s) = \{(x(s), y(s)) : u(x(s), y(s)) = l\}.$$

By differentiating the equality $u(x(s), y(s)) = l$ with respect to s, we obtain

$$u_x x'(s) + u_y y'(s) = 0. \tag{32}$$

Hence the vectors $(x'(s), y'(s))$ and $(-u_y, u_x)$ are collinear, or equivalently for some λ we have

$$\begin{cases} x'(s) = -\lambda u_y, \\ y'(s) = \lambda u_x. \end{cases} \tag{33}$$

Note that since $|\mathbf{c}'(s)| = 1$, from (33) we get $\lambda = 1/|\nabla u|$ (supposing $|\nabla u| \neq 0$). If differentiating again (32) with respect to s we obtain

$$u_{xx}(x'(s))^2 + u_{yy}(y'(s))^2 + 2u_{xy}x'(s)y'(s) + u_x x''(s) + u_y y''(s) = 0.$$

Plugging (33) into the above equality leads to

$$\lambda^2[u_{xx}(u_y)^2 + u_{yy}(u_x)^2 - 2u_{xy}u_y u_x] + \frac{1}{\lambda}[y'(s)x''(s) - x'(s)y''(s)] = 0.$$

By (31) we can deduce the curvature expression as

$$\kappa(s) = \frac{|\mathbf{c}'(s) \times \mathbf{c}''(s)|}{|\mathbf{c}'(s)|^3} = x'(s)y''(s) - x''(s)y'(s) = \frac{u_{xx}(u_y)^2 + u_{yy}(u_x)^2 - 2u_{xy}u_y u_x}{|\nabla u|^3}.$$

Denoting $f \doteq |\nabla u| = \sqrt{(u_x)^2 + (u_y)^2}$, we have

$$\begin{aligned}
\nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) &= \frac{\partial}{\partial x}(\frac{1}{f}u_x) + \frac{\partial}{\partial y}(\frac{1}{f}u_y) \\
&= \frac{\partial}{\partial x}(\frac{1}{f})u_x + \frac{1}{f}u_{xx} + \frac{\partial}{\partial y}(\frac{1}{f})u_y + \frac{1}{f}u_{yy} \\
&= -\frac{1}{f^2}f_x u_x - \frac{1}{f^2}f_y u_y + \frac{1}{f}(u_{xx} + u_{yy}) \\
&= -\frac{1}{f^2}\Big[\frac{1}{f}(u_x u_{xx} + u_y u_{yx})\Big]u_x - \frac{1}{f^2}\Big[\frac{1}{f}(u_x u_{xy} + u_y u_{yy})\Big]u_y + \frac{1}{f}(u_{xx} + u_{yy}) \\
&= -\frac{1}{f^3}\Big\{(u_x)^2 u_{xx} + (u_y)^2 u_{yy} + 2u_x u_y u_{xy} - \Big[(u_x)^2 + (u_y)^2\Big](u_{xx} + u_{yy})\Big\} \\
&= \frac{u_{xx}(u_y)^2 + u_{yy}(u_x)^2 - 2u_{xy}u_y u_x}{|\nabla u|^3} \\
&= \kappa(s),
\end{aligned}$$

thus getting the curvature expression (6).

Since the above derivations only consider the case of level curves for a 2D function $u(x, y)$, here we give some remarks on the curvature expression (6) in high dimensional spaces. For a level surface defined in 3D space, the curvature expression (6) at point p amounts to the mean curvature of this surface:

$$H = \frac{1}{2}\nabla \cdot \mathbf{N},$$

where $\mathbf{N}$ is a unit normal of the surface (see Chan and Shen, 2005, chap. 2.1.2). Formally, the mean curvature is defined as the average of the principal curvatures (Spivak, 1999, vol. 3, chap. 2): $H = (\kappa_1 + \kappa_2)/2$, where κ_1 and κ_2 are two principal curvatures. In this case, the Gaussian curvature is given by $K = \kappa_1 \cdot \kappa_2$. More generally (Spivak, 1999, vol. 4, chap. 7), for a $(d-1)$-dimensional level hypersurface embedded in R^d the mean curvature is given as $H = (\kappa_1 + \cdots + \kappa_{d-1})/(d-1)$ in terms of principal curvatures. More abstractly, the mean curvature is the trace of the second fundamental form divided by $d-1$ (or equivalently the shape operator or Weingarten map). The shape operator s (Lee, 1997, chap. 8) is an extrinsic curvature, and the Gaussian curvature is given by the determinant of s. Mean curvature is closely related to the first variation of surface area, in particular a minimal surface such as a soap film, has mean curvature zero and a soap bubble has constant mean curvature. Unlike Gauss curvature, the mean curvature is extrinsic and depends on the embedding, for instance, a cylinder and a plane are locally isometric but the mean curvature of a plane is zero while that of a cylinder is nonzero (see http://en.wikipedia.org/wiki/Curvature). One can also refer to Ambrosio and Masnou (2003) for the description of this high dimensional representation.

Appendix B: PDEs Derived by Calculus of Variations

We present the following derivations of the Euler-Lagrange PDEs by calculus of variations (van Brunt, 2004). Note that the variation operator δ acts much like a differentiation

operator. First we list some expressions about δ which are useful in the following derivations (where $\mathbf{F}$ is a d-dimensional differentiable vector field):

$$\begin{aligned}
\delta(\nabla u) &= \nabla(\delta u),\\
\delta(\nabla\cdot\mathbf{F}) &= \delta\Big(\sum_{i=1}^{d}\frac{\partial F^{(i)}}{\partial x^{(i)}}\Big)=\sum_{i=1}^{d}\delta\Big(\frac{\partial F^{(i)}}{\partial x^{(i)}}\Big)=\sum_{i=1}^{d}\frac{\partial(\delta F)^{(i)}}{\partial x^{(i)}}=\nabla\cdot\delta\mathbf{F},\\
\delta(|\nabla u|^2) &= \delta\Big[\sum_{i=1}^{d}\Big(\frac{\partial u}{\partial x^{(i)}}\Big)^2\Big]=\sum_{i=1}^{d}2\frac{\partial u}{\partial x^{(i)}}\delta\Big(\frac{\partial u}{\partial x^{(i)}}\Big)=2\langle\nabla u,\delta\nabla u\rangle=2\langle\nabla u,\nabla\delta u\rangle,\\
\delta(|\nabla u|) &= \delta\Big\{\Big[\sum_{i=1}^{d}\Big(\frac{\partial u}{\partial x^{(i)}}\Big)^2\Big]^{1/2}\Big\}=\frac{1}{2}\Big[\sum_{i=1}^{d}\Big(\frac{\partial u}{\partial x^{(i)}}\Big)^2\Big]^{-1/2}\delta\Big[\sum_{i=1}^{d}\Big(\frac{\partial u}{\partial x^{(i)}}\Big)^2\Big]\\
&= \frac{1}{2}\frac{1}{|\nabla u|}2\langle\nabla u,\nabla\delta u\rangle=\Big\langle\frac{\nabla u}{|\nabla u|},\nabla\delta u\Big\rangle,\\
\delta\Big(\frac{1}{|\nabla u|}\Big) &= \delta\Big\{\Big[\sum_{i=1}^{d}\Big(\frac{\partial u}{\partial x^{(i)}}\Big)^2\Big]^{-1/2}\Big\}=-\frac{1}{2}\Big[\sum_{i=1}^{d}\Big(\frac{\partial u}{\partial x^{(i)}}\Big)^2\Big]^{-3/2}\delta\Big[\sum_{i=1}^{d}\Big(\frac{\partial u}{\partial x^{(i)}}\Big)^2\Big]\\
&= -\frac{1}{2}\frac{1}{|\nabla u|^3}2\langle\nabla u,\nabla\delta u\rangle=-\frac{1}{|\nabla u|^3}\langle\nabla u,\nabla\delta u\rangle.
\end{aligned}$$

Proof of (9)⇒(10). Suppose $u:R^d\to R$ is a differentiable function. The first variation, $E_{LR}\to E_{LR}+\delta E_{LR}$, under $u\to u+\delta u$ is given by

$$\begin{aligned}
\delta E_{LR} &= \delta\Big\{\int_\Omega\Big[(u-y)^2+\lambda|\nabla u|^2\Big]d\mathbf{x}\Big\}\\
&= \int_\Omega\Big\{\delta\Big[(u-y)^2\Big]+\lambda\delta\Big(|\nabla u|^2\Big)\Big\}d\mathbf{x}\\
&= \int_\Omega\Big[2(u-y)\delta u+2\lambda\langle\nabla u,\nabla\delta u\rangle\Big]d\mathbf{x}\\
&= 2\Big[\int_\Omega(u-y)\delta u d\mathbf{x}+\int_{\partial\Omega}\lambda\nabla u\delta u\cdot\mathbf{n}dS-\int_\Omega\lambda(\nabla\cdot\nabla u)\delta u d\mathbf{x}\Big] \qquad (34)\\
&= 2\Big[\int_\Omega(u-y)\delta u d\mathbf{x}+\int_{\partial\Omega}\lambda\frac{\partial u}{\partial\mathbf{n}}\delta u dS-\int_\Omega\lambda(\nabla\cdot\nabla u)\delta u d\mathbf{x}\Big] \qquad (35)\\
&= 2\int_\Omega\Big[(u-y)-\lambda\Delta u\Big]\delta u d\mathbf{x}. \qquad (36)
\end{aligned}$$

Here $\nabla\cdot$ is the divergence operator, Δ is the Laplacian operator, and $\mathbf{n}$ denotes the outer normal along the boundary $\partial\Omega$. The equation (34) is obtained based on the Gauss-Green divergence theorem in vector calculus (Spiegel and Lipschutz, 2009) (which is a special case of the more general Stokes' theorem):

$$\int_V(\nabla\cdot\mathbf{F})dV=\int_S(\mathbf{F}\cdot\mathbf{n})dS,$$

where V is a subset of R^d (in the case of $d=3$, V represents a volume in 3D space) which is compact and has a piecewise smooth boundary S (also indicated with $\partial V=S$), $\mathbf{F}$ is a

continuously differentiable vector field, and $\mathbf{n}$ is the outward pointing unit normal field of the boundary ∂V. In fact, we use the following corollary of the divergence theorem when applied to the product of a scalar function g (that is δu in our context) and a vector field $\mathbf{F}$ (∇u in our context):

$$\int_V [\mathbf{F} \cdot (\nabla g) + g(\nabla \cdot \mathbf{F})] dV = \int_S (g\mathbf{F} \cdot \mathbf{n}) dS.$$

Then integration by parts implies (34). The equation (35) is written with the directional derivative notation $\partial u/\partial \mathbf{n} \doteq \nabla u \cdot \mathbf{n} = \langle \nabla u, \mathbf{n} \rangle$. The last equation (36) is due to the assumption of *natural boundary conditions*

$$\frac{\partial u}{\partial \mathbf{n}}|_{\partial\Omega} = 0.$$

According to the *fundamental lemma of calculus of variations*, the integrand part in parentheses is equal to zero because δu is an arbitrary function. Hence we obtain (10).

Proof of (11)⇒(12). The first variation, $E_{TV} \to E_{TV} + \delta E_{TV}$, under $u \to u + \delta u$ is given by

$$\begin{aligned}
\delta E_{TV} &= \delta\Big\{ \int_\Omega \Big[\frac{1}{2}(u-y)^2 + \lambda|\nabla u|\Big] d\mathbf{x}\Big\} \\
&= \int_\Omega \Big\{(u-y)\delta u + \lambda\delta(|\nabla u|)\Big\} d\mathbf{x} \\
&= \int_\Omega \Big[(u-y)\delta u + \lambda\Big\langle \frac{\nabla u}{|\nabla u|}, \nabla\delta u\Big\rangle\Big] d\mathbf{x} \\
&= \int_\Omega (u-y)\delta u d\mathbf{x} + \int_{\partial\Omega} \lambda \frac{1}{|\nabla u|}\frac{\partial u}{\partial \mathbf{n}}\delta u dS - \int_\Omega \lambda\Big(\nabla \cdot \frac{\nabla u}{|\nabla u|}\Big)\delta u d\mathbf{x} \qquad (37) \\
&= \int_\Omega \Big[(u-y) - \lambda\nabla \cdot \frac{\nabla u}{|\nabla u|}\Big]\delta u d\mathbf{x}. \qquad (38)
\end{aligned}$$

Again the integration term over the boundary $\partial\Omega$ in (37) can be removed by the natural boundary conditions. By the fundamental lemma of calculus of variations, the integrand part in parentheses of (38) must equal to zero. Thus we get (12).

Proof of (14)⇒(15). The original derivation comes from Chan et al. (2002). Let $f(\kappa) = a + b\kappa^2$ and the elastica regularization term be

$$R(u) = \int_\Omega f(\kappa)|\nabla u| d\mathbf{x}.$$

We need to prove that the first variation, $R(u) \to R(u) + \delta R(u)$, under $u \to u + \delta u$ is given by

$$\delta R(u) = \int_\Omega -\nabla \cdot \mathbf{V}(u)\delta u d\mathbf{x},$$

where $\mathbf{V}(u)$ is a flux field defined as

$$\mathbf{V}(u) = f(\kappa)\mathbf{N} - \frac{\mathbf{T}}{|\nabla u|}\frac{\partial(f'(\kappa)|\nabla u|)}{\partial \mathbf{T}}.$$

Here $\mathbf{N}$ is the ascending normal field $\nabla u/|\nabla u|$, and $\mathbf{T}$ is the tangent field defined as $\mathbf{T} = \mathbf{N}^{\perp}$. Note that the exact orientation of $\mathbf{T}$ does not matter due to the coupling of $\mathbf{T}$ and $\partial/\partial\mathbf{T}$ in the expression. Since the curvature κ is a function of u, by variational rules we have

$$\begin{aligned}
\delta R(u) &= \delta\Big\{\int_{\Omega} f(\kappa)|\nabla u| d\mathbf{x}\Big\} \\
&= \int_{\Omega}\Big\{|\nabla u|\delta\Big[f(\kappa)\Big] + f(\kappa)\delta(|\nabla u|)\Big\} d\mathbf{x} \\
&= \int_{\Omega}\Big\{|\nabla u| f'(\kappa)\delta\kappa + f(\kappa)\Big\langle\frac{\nabla u}{|\nabla u|}, \nabla\delta u\Big\rangle\Big\} d\mathbf{x} \\
&= \int_{\Omega}|\nabla u| f'(\kappa)\delta\kappa d\mathbf{x} + \int_{\partial\Omega}\frac{f(\kappa)}{|\nabla u|}\frac{\partial u}{\partial\mathbf{n}}\delta u dS - \int_{\Omega} f(\kappa)\Big(\nabla\cdot\frac{\nabla u}{|\nabla u|}\Big)\delta u d\mathbf{x} \qquad (39) \\
&= \int_{\Omega}\Big\{|\nabla u| f'(\kappa)\delta\kappa - f(\kappa)\Big(\nabla\cdot\frac{\nabla u}{|\nabla u|}\Big)\delta u\Big\} d\mathbf{x} \\
&= \int_{\Omega}\Big\{|\nabla u| f'(\kappa)\delta\kappa - \Big[\nabla\cdot\big(f(\kappa)\mathbf{N}\big)\Big]\delta u\Big\} d\mathbf{x}.
\end{aligned}$$

Here the integration term over the boundary $\partial\Omega$ in (39) can be removed by the natural boundary conditions. The variation of curvature $\kappa = \nabla\cdot\mathbf{N}$ is a function of δu, which can be further written as

$$\begin{aligned}
\delta\kappa &= \delta(\nabla\cdot\mathbf{N}) \\
&= \nabla\cdot\delta\mathbf{N} \\
&= \nabla\cdot\delta\Big(\frac{\nabla u}{|\nabla u|}\Big) \\
&= \nabla\cdot\Big[\frac{1}{|\nabla u|}\delta(\nabla u) + \nabla u\delta\Big(\frac{1}{|\nabla u|}\Big)\Big] \\
&= \nabla\cdot\Big[\frac{1}{|\nabla u|}\nabla(\delta u) - \nabla u\Big(\frac{1}{|\nabla u|^3}\langle\nabla u, \nabla(\delta u)\rangle\Big)\Big] \\
&= \nabla\cdot\Big[\frac{1}{|\nabla u|}\nabla(\delta u) - \frac{1}{|\nabla u|}\mathbf{N}\langle\mathbf{N}, \nabla(\delta u)\rangle\Big] \\
&= \nabla\cdot\Big[\frac{1}{|\nabla u|}(\mathbf{I} - \mathbf{N}\otimes\mathbf{N})\nabla(\delta u)\Big] \\
&= \nabla\cdot\Big[\frac{1}{|\nabla u|}P_{\mathbf{T}}(\nabla(\delta u))\Big].
\end{aligned}$$

Here $\mathbf{I}$ denotes the identity transform, $P_{\mathbf{N}} \doteq \mathbf{N}\otimes\mathbf{N}$ is the orthogonal projection onto the ascending normal direction of u, and $P_{\mathbf{T}} \doteq \mathbf{I} - \mathbf{N}\otimes\mathbf{N} = \mathbf{T}\otimes\mathbf{T}$ is the orthogonal projection onto the tangent direction of u. Therefore by the Gauss-Green divergence theorem we have

$$\begin{aligned}
&\int_{\Omega}|\nabla u| f'(\kappa)\delta\kappa d\mathbf{x} \\
=\ &\int_{\Omega} f'(\kappa)|\nabla u|\Big\{\nabla\cdot\Big[\frac{1}{|\nabla u|}P_{\mathbf{T}}(\nabla(\delta u))\Big]\Big\} d\mathbf{x} \\
=\ &\int_{\partial\Omega} f'(\kappa)P_{\mathbf{T}}(\nabla(\delta u))\cdot\mathbf{n} dS - \int_{\Omega}\Big\langle\nabla\Big[f'(\kappa)|\nabla u|\Big], \frac{1}{|\nabla u|}P_{\mathbf{T}}(\nabla(\delta u))\Big\rangle d\mathbf{x}
\end{aligned}$$

$$
\begin{aligned}
&= -\int_\Omega \Big\langle \nabla\Big[f'(\kappa)|\nabla u|\Big], \frac{1}{|\nabla u|} P_{\mathbf{T}}(\nabla(\delta u))\Big\rangle d\mathbf{x} \\
&= -\int_\Omega \Big\langle \frac{1}{|\nabla u|} P_{\mathbf{T}}\Big\{\nabla\Big[f'(\kappa)|\nabla u|\Big]\Big\}, \nabla(\delta u)\Big\rangle d\mathbf{x} \qquad (40) \\
&= -\int_{\partial\Omega} \frac{1}{|\nabla u|} P_{\mathbf{T}}\Big\{\nabla\Big[f'(\kappa)|\nabla u|\Big]\Big\}\delta u \cdot \mathbf{n} dS + \int_\Omega \nabla\cdot \frac{1}{|\nabla u|} P_{\mathbf{T}}\Big\{\nabla\Big[f'(\kappa)|\nabla u|\Big]\Big\}\delta u d\mathbf{x} \\
&= \int_\Omega \nabla\cdot \frac{1}{|\nabla u|} P_{\mathbf{T}}\Big\{\nabla\Big[f'(\kappa)|\nabla u|\Big]\Big\}\delta u d\mathbf{x},
\end{aligned}
$$

where proper natural boundary conditions are imposed to remove the integrations over the boundary $\partial\Omega$, and the equation (40) is given by the symmetry property of the projection operator $P_{\mathbf{T}}$ in an inner product. Finally, using the definition of directional derivative $P_{\mathbf{T}}(\nabla f) = \mathbf{T}(\partial f/\partial \mathbf{T})$, we complete the derivations of (14)⇒(15) by

$$
\begin{aligned}
\delta R(u) &= \int_\Omega \Big\{|\nabla u| f'(\kappa)\delta\kappa - \Big[\nabla\cdot(f(\kappa)\mathbf{N})\Big]\delta u\Big\} d\mathbf{x} \\
&= \int_\Omega \Big\{\nabla\cdot \frac{1}{|\nabla u|} P_{\mathbf{T}}\Big\{\nabla\Big[f'(\kappa)|\nabla u|\Big]\Big\} - \nabla\cdot(f(\kappa)\mathbf{N})\Big\}\delta u d\mathbf{x} \\
&= -\int_\Omega \nabla\cdot\Big\{f(\kappa)\mathbf{N} - \frac{1}{|\nabla u|} P_{\mathbf{T}}\Big\{\nabla\Big[f'(\kappa)|\nabla u|\Big]\Big\}\Big\}\delta u d\mathbf{x} \\
&= -\int_\Omega \nabla\cdot\Big\{f(\kappa)\mathbf{N} - \frac{1}{|\nabla u|}\frac{\partial(f'(\kappa)|\nabla u|)}{\partial \mathbf{T}}\mathbf{T}\Big\}\delta u d\mathbf{x} \\
&= -\int_\Omega \nabla\cdot \mathbf{V}\delta u d\mathbf{x}.
\end{aligned}
$$

Appendix C: Expressions in Terms of RBF Approximations

The following gives some useful expressions about Laplacian, Hessian, and curvature of $u(\mathbf{x})$ in terms of RBF approximations $u(\mathbf{x}) = \sum_{i=1}^n w_i\phi_i(\mathbf{x})$, where $\phi_i(\mathbf{x}) = \exp(-c|\mathbf{x} - \mathbf{x_i}|^2/2)$.

Proof of (18) for Laplacian:

$$
\begin{aligned}
\frac{\partial^2\phi_k}{\partial x^{(i)}\partial x^{(i)}} &= \frac{\partial}{\partial x^{(i)}}\Big(\frac{\partial\phi_k}{\partial x^{(i)}}\Big) \\
&= \frac{\partial}{\partial x^{(i)}}\Big[-c(x^{(i)} - x_k^{(i)})\phi_k\Big] \\
&= -c(x^{(i)} - x_k^{(i)})\frac{\partial\phi_k}{\partial x^{(i)}} - c\phi_k\frac{\partial(x^{(i)} - x_k^{(i)})}{\partial x^{(i)}} \\
&= -c(x^{(i)} - x_k^{(i)})[-c(x^{(i)} - x_k^{(i)})\phi_k] - c\phi_k \\
&= c[c(x^{(i)} - x_k^{(i)})^2 - 1]\phi_k. \\
\Delta\phi_k &= \sum_{i=1}^d \frac{\partial^2\phi_k}{\partial x^{(i)}\partial x^{(i)}} = c(c|x - x_k|^2 - d)\phi_k.
\end{aligned}
$$

Proof of (19) for Hessian:

$$
(\text{for } i \neq j) \quad \frac{\partial^2\phi_k}{\partial x^{(i)}\partial x^{(j)}} = \frac{\partial}{\partial x^{(j)}}[-c(x^{(i)} - x_k^{(i)})\phi_k]
$$

$$
\begin{aligned}
& = -c(x^{(i)} - x_k^{(i)})[-c(x^{(j)} - x_k^{(j)})\phi_k] \\
& = c^2(x^{(i)} - x_k^{(i)})(x^{(j)} - x_k^{(j)})\phi_k. \\
\text{(for } i = j) \quad \frac{\partial^2 \phi_k}{\partial x^{(i)} \partial x^{(j)}} & = c[c(x^{(i)} - x_k^{(i)})^2 - 1]\phi_k. \\
\mathbf{H}(\phi_k) & = c^2(\mathbf{x} - \mathbf{x}_k)(\mathbf{x} - \mathbf{x}_k)^T \phi_k - c\phi_k \mathbf{I}.
\end{aligned}
$$

Proof of (21) for Hessian in terms of notation $\Phi \doteq \sum_{i=1}^n w_i(\mathbf{x} - \mathbf{x}_i)(\mathbf{x} - \mathbf{x}_i)^T \phi_i$:

$$
\begin{aligned}
\mathbf{H}(u) & = \sum_{i=1}^n w_i \mathbf{H}(\phi_i) \\
& = \sum_{i=1}^n w_i[-c\phi_i \mathbf{I} + c^2(\mathbf{x} - \mathbf{x}_i)(\mathbf{x} - \mathbf{x}_i)^T \phi_i] \\
& = -c \sum_{i=1}^n w_i \phi_i \mathbf{I} + c^2 \sum_{i=1}^n w_i(\mathbf{x} - \mathbf{x}_i)(\mathbf{x} - \mathbf{x}_i)^T \phi_i \\
& = -c\Big(\sum_{i=1}^n w_i \phi_i\Big)\mathbf{I} + c^2 \Phi.
\end{aligned}
$$

To prove (23) and others derivations involving gradients in Appendix D, here we list some useful expressions (notice that we do not distinguish $\mathbf{H}(u)$ from $\mathbf{H}(u)^T$ due to symmetry):

$$
\begin{aligned}
\nabla(|\nabla u|^2) & = \nabla\Big[\sum_{i=1}^d \Big(\frac{\partial u}{\partial x^{(i)}}\Big)^2\Big] = \sum_{i=1}^d 2\frac{\partial u}{\partial x^{(i)}} \nabla\Big(\frac{\partial u}{\partial x^{(i)}}\Big) \\
& = \sum_{i=1}^d 2\frac{\partial u}{\partial x^{(i)}} \begin{pmatrix} \frac{\partial^2 u}{\partial x^{(i)} \partial x^{(1)}} \\ \cdots \\ \frac{\partial^2 u}{\partial x^{(i)} \partial x^{(d)}} \end{pmatrix} = 2\mathbf{H}(u)\nabla u, \\
\nabla(|\nabla u|) & = \nabla\Big\{\Big[\sum_{i=1}^d \Big(\frac{\partial u}{\partial x^{(i)}}\Big)^2\Big]^{1/2}\Big\} = \frac{1}{2}\Big[\sum_{i=1}^d \Big(\frac{\partial u}{\partial x^{(i)}}\Big)^2\Big]^{-1/2} \nabla\Big[\sum_{i=1}^d \Big(\frac{\partial u}{\partial x^{(i)}}\Big)^2\Big] \\
& = \frac{1}{2|\nabla u|} 2\mathbf{H}(u)\nabla u = \frac{1}{|\nabla u|}\mathbf{H}(u)\nabla u, \\
\nabla\Big(\frac{1}{|\nabla u|}\Big) & = \nabla\Big\{\Big[\sum_{i=1}^d \Big(\frac{\partial u}{\partial x^{(i)}}\Big)^2\Big]^{-1/2}\Big\} = -\frac{1}{2}\Big[\sum_{i=1}^d \Big(\frac{\partial u}{\partial x^{(i)}}\Big)^2\Big]^{-3/2} \nabla\Big[\sum_{i=1}^d \Big(\frac{\partial u}{\partial x^{(i)}}\Big)^2\Big] \\
& = -\frac{1}{2|\nabla u|^3} 2\mathbf{H}(u)\nabla u = -\frac{1}{|\nabla u|^3}\mathbf{H}(u)\nabla u, \\
\nabla\Big(\frac{1}{|\nabla u|^3}\Big) & = \nabla\Big\{\Big[\sum_{i=1}^d \Big(\frac{\partial u}{\partial x^{(i)}}\Big)^2\Big]^{-3/2}\Big\} = -\frac{3}{2}\Big[\sum_{i=1}^d \Big(\frac{\partial u}{\partial x^{(i)}}\Big)^2\Big]^{-5/2} \nabla\Big[\sum_{i=1}^d \Big(\frac{\partial u}{\partial x^{(i)}}\Big)^2\Big] \\
& = -\frac{3}{2|\nabla u|^5} 2\mathbf{H}(u)\nabla u = -\frac{3}{|\nabla u|^5}\mathbf{H}(u)\nabla u.
\end{aligned}
$$

Proof of (23) for curvature:

$$
\kappa \doteq \nabla \cdot \Big(\frac{\nabla u}{|\nabla u|}\Big)
$$

$$
\begin{aligned}
&= \frac{1}{|\nabla u|}\nabla\cdot\nabla u + \nabla\Big(\frac{1}{|\nabla u|}\Big)\cdot\nabla u \\
&= \frac{1}{|\nabla u|}\Delta u - \frac{1}{|\nabla u|^3}\mathbf{H}(u)\nabla u\cdot\nabla u \\
&= \frac{1}{|\nabla u|}\Big(\Delta u - \frac{\nabla u^T\mathbf{H}(u)\nabla u}{\nabla u^T\nabla u}\Big) \\
&= \frac{1}{|-c\mathbf{g}|}\left\{c\sum_i w_i(c|\mathbf{x}-\mathbf{x}_i|^2-d)\phi_i - \frac{(-c\mathbf{g})^T(c^2\Phi - c(\sum_i w_i\phi_i)\mathbf{I})(-c\mathbf{g})}{(-c\mathbf{g})^T(-c\mathbf{g})}\right\} \\
&= \frac{1}{|\mathbf{g}|}\left\{\sum_i w_i(c|\mathbf{x}-\mathbf{x}_i|^2-d)\phi_i - \frac{\mathbf{g}^T(c\Phi - (\sum_i w_i\phi_i)\mathbf{I})\mathbf{g}}{\mathbf{g}^T\mathbf{g}}\right\} \\
&= \frac{1}{|\mathbf{g}|}\left\{\sum_i w_i(c|\mathbf{x}-\mathbf{x}_i|^2-d+1)\phi_i - c\frac{\mathbf{g}^T\Phi\mathbf{g}}{\mathbf{g}^T\mathbf{g}}\right\}.
\end{aligned}
$$

Appendix D: Expansion of $\nabla\cdot\mathbf{V}$ in Gradient Descent Time Marching

Here we give the derivations of the expansion (26) of $\nabla\cdot\mathbf{V}$ in Gradient Descent Time Marching. For simpler notations we define

$$\alpha \doteq \nabla u^T\mathbf{H}(u)\nabla u, \quad \beta \doteq \nabla u^T\mathbf{H}(u)^2\nabla u, \quad \gamma \doteq \nabla u^T\mathbf{H}(u)^3\nabla u.$$

By definition (16)

$$
\begin{aligned}
\mathbf{V}(u) &\doteq f(\kappa)\mathbf{N} - \frac{\mathbf{T}}{|\nabla u|}\frac{\partial(f'(\kappa)|\nabla u|)}{\partial\mathbf{T}} \\
&= f(\kappa)\mathbf{N} - \frac{1}{|\nabla u|}\nabla(f'(\kappa)|\nabla u|) + \frac{1}{|\nabla u|^3}\nabla u\langle\nabla u, \nabla\left(f'(\kappa)|\nabla u|\right)\rangle \\
&= (1+b\kappa^2)\mathbf{N} - \frac{1}{|\nabla u|}\nabla(2b\kappa|\nabla u|) + \frac{1}{|\nabla u|^3}\nabla u\langle\nabla u, \nabla\left(2b\kappa|\nabla u|\right)\rangle,
\end{aligned}
$$

we have

$$\nabla\cdot\mathbf{V} = \nabla\cdot[(1+b\kappa^2)\mathbf{N}] - 2b\nabla\cdot\Big[\frac{1}{|\nabla u|}\nabla(\kappa|\nabla u|)\Big] + 2b\nabla\cdot\Big\{\frac{1}{|\nabla u|^3}\nabla u\Big[\nabla u^T\nabla(\kappa|\nabla u|)\Big]\Big\}. \quad (41)$$

Then we show the following derivations for the three parts on the right side of (41).

Part 1: The first term can be expanded as

$$\nabla\cdot\mathbf{N} + b\nabla\cdot(\kappa^2\mathbf{N}) = \kappa + b[\nabla(\kappa^2)\cdot\mathbf{N} + \kappa^2\nabla\cdot\mathbf{N}] = \kappa + b(2\kappa\nabla\kappa\cdot\mathbf{N} + \kappa^3),$$

where $\nabla\kappa$ can be further written as

$$
\begin{aligned}
\nabla\kappa &= \nabla\Big[\nabla\cdot\Big(\frac{\nabla u}{|\nabla u|}\Big)\Big] \\
&= \nabla\Big[\nabla\Big(\frac{1}{|\nabla u|}\Big)\cdot\nabla u + \frac{1}{|\nabla u|}\Delta u\Big]
\end{aligned}
$$

$$
\begin{aligned}
&= \cancel{\nabla\Big[\nabla\Big(\frac{1}{|\nabla u|}\Big)\Big]\cdot\nabla u} + \nabla(\nabla u)\cdot\nabla\Big(\frac{1}{|\nabla u|}\Big) + \nabla\Big(\frac{1}{|\nabla u|}\Big)\Delta u + \cancel{\frac{1}{|\nabla u|}\nabla(\Delta u)}\\
&\approx \mathbf{H}(u)\nabla\Big(\frac{1}{|\nabla u|}\Big) + \Delta u\nabla\Big(\frac{1}{|\nabla u|}\Big)\\
&= -\frac{1}{|\nabla u|^3}[\mathbf{H}(u)^2\nabla u + \Delta u\mathbf{H}(u)\nabla u].
\end{aligned}
$$

Here the third equality is obtained by the formula for the gradient of a dot product

$$
\begin{aligned}
\nabla(\mathbf{a}\cdot\mathbf{b}) &= (\nabla\mathbf{a})\cdot\mathbf{b} + (\nabla\mathbf{b})\cdot\mathbf{a}\\
&= \begin{pmatrix} \frac{\partial a_1}{\partial x_1} & \cdots & \frac{\partial a_d}{\partial x_1} \\ \cdots & \cdots & \cdots \\ \frac{\partial a_1}{\partial x_d} & \cdots & \frac{\partial a_d}{\partial x_d} \end{pmatrix}\mathbf{b} + \begin{pmatrix} \frac{\partial b_1}{\partial x_1} & \cdots & \frac{\partial b_d}{\partial x_1} \\ \cdots & \cdots & \cdots \\ \frac{\partial b_1}{\partial x_d} & \cdots & \frac{\partial b_d}{\partial x_d} \end{pmatrix}\mathbf{a},
\end{aligned}
$$

and we omit the third order derivatives by notation $\not{\ }$ for easier calculations. Therefore, the first term on the right side of (41) can be written as

$$
\begin{aligned}
\nabla\cdot((1+b\kappa^2)\mathbf{N}) &= \kappa + b\kappa^3 - \frac{2b\kappa}{|\nabla u|^4}[\nabla u^T\mathbf{H}(u)^2\nabla u + \Delta u\nabla u^T\mathbf{H}(u)\nabla u]\\
&= \kappa + b\kappa^3 - \frac{2b\kappa}{|\nabla u|^4}(\alpha\Delta u + \beta).
\end{aligned}
$$

Part 2: The second term on the right side of (41) can be expanded as

$$
\begin{aligned}
&\nabla\cdot\Big[\frac{1}{|\nabla u|}\nabla(\kappa|\nabla u|)\Big]\\
=\ &\nabla\Big(\frac{1}{|\nabla u|}\Big)\cdot\nabla(\kappa|\nabla u|) + \frac{1}{|\nabla u|}\nabla\cdot\Big[\nabla(\kappa|\nabla u|)\Big]\\
=\ &-\frac{1}{|\nabla u|^3}\mathbf{H}(u)\nabla u\cdot\Big[|\nabla u|\nabla\kappa + \kappa\nabla(|\nabla u|)\Big] + \frac{1}{|\nabla u|}\Big\{\nabla\cdot\Big[|\nabla u|\nabla\kappa + \kappa\nabla(|\nabla u|)\Big]\Big\}\\
=\ &-\frac{1}{|\nabla u|^2}\nabla u^T\mathbf{H}(u)\nabla\kappa - \frac{\kappa}{|\nabla u|^3}\nabla u^T\mathbf{H}(u)\nabla(|\nabla u|)\\
&+\frac{1}{|\nabla u|}\Big[\nabla(|\nabla u|)\cdot\nabla\kappa + \cancel{|\nabla u|\nabla\cdot\nabla\kappa} + \nabla\kappa\cdot\nabla(|\nabla u|) + \cancel{\kappa\nabla\cdot\nabla(|\nabla u|)}\Big]\\
\approx\ &-\frac{1}{|\nabla u|^2}\nabla u^T\mathbf{H}(u)\nabla\kappa - \frac{\kappa}{|\nabla u|^3}\nabla u^T\mathbf{H}(u)\Big[\frac{1}{|\nabla u|}\mathbf{H}(u)\nabla u\Big] + \frac{2}{|\nabla u|^2}\nabla u^T\mathbf{H}(u)\nabla\kappa\\
=\ &\frac{1}{|\nabla u|^2}\nabla u^T\mathbf{H}(u)\nabla\kappa - \frac{\kappa}{|\nabla u|^4}\nabla u^T\mathbf{H}(u)^2\nabla u\\
=\ &\frac{1}{|\nabla u|^2}\nabla u^T\mathbf{H}(u)\Big\{-\frac{1}{|\nabla u|^3}[\mathbf{H}(u)^2\nabla u + \Delta u\mathbf{H}(u)\nabla u]\Big\} - \frac{\kappa}{|\nabla u|^4}\beta\\
=\ &-\Big(\frac{\Delta u}{|\nabla u|^5} + \frac{\kappa}{|\nabla u|^4}\Big)\beta - \frac{1}{|\nabla u|^5}\gamma.
\end{aligned}
$$

Part 3: Finally we consider the third term on the right side of (41). With notation $\mathbf{v} \doteq \nabla(\kappa|\nabla u|)$, we have

$$
\nabla\cdot\Big\{\frac{1}{|\nabla u|^3}\nabla u\Big[\nabla u^T\nabla(\kappa|\nabla u|)\Big]\Big\}
$$

$$
\begin{aligned}
&= \nabla\cdot\Big[\frac{1}{|\nabla u|^3}\nabla u(\nabla u\cdot\mathbf{v})\Big]\\
&= \nabla\Big(\frac{\nabla u\cdot\mathbf{v}}{|\nabla u|^3}\Big)\cdot\nabla u+\Big(\frac{\nabla u\cdot\mathbf{v}}{|\nabla u|^3}\Big)\Delta u\\
&= \Big[\nabla\Big(\frac{1}{|\nabla u|^3}\Big)(\nabla u\cdot\mathbf{v})+\frac{1}{\cancel{|\nabla u|^3}}\cancel{\nabla(\nabla u\cdot\mathbf{v})}\Big]\cdot\nabla u+\Big(\frac{\nabla u\cdot\mathbf{v}}{|\nabla u|^3}\Big)\Delta u\\
&\approx \Big[\nabla\Big(\frac{1}{|\nabla u|^3}\Big)\cdot\nabla u+\frac{\Delta u}{|\nabla u|^3}\Big](\nabla u\cdot\mathbf{v})\\
&= \Big(\frac{\Delta u}{|\nabla u|^3}-\frac{3}{|\nabla u|^5}\alpha\Big)(\nabla u\cdot\mathbf{v}).
\end{aligned}
$$

Because

$$
\begin{aligned}
\nabla u\cdot\mathbf{v} &= \nabla u\cdot[\nabla(\kappa|\nabla u|)]\\
&= \nabla u\cdot(|\nabla u|\nabla\kappa+\kappa\nabla(|\nabla u|))\\
&= |\nabla u|\nabla u\cdot\nabla\kappa+\kappa\nabla u\cdot\nabla(|\nabla u|)\\
&= |\nabla u|\nabla u\cdot\Big\{-\frac{1}{|\nabla u|^3}[\mathbf{H}(u)^2\nabla u+\Delta u\mathbf{H}(u)\nabla u]\Big\}+\kappa\nabla u\cdot\Big(\frac{1}{|\nabla u|}\mathbf{H}(u)\nabla u\Big)\\
&= \Big(\frac{\kappa}{|\nabla u|}-\frac{\Delta u}{|\nabla u|^2}\Big)\alpha-\frac{1}{|\nabla u|^2}\beta,
\end{aligned}
$$

we obtain the expansion of the third term on the right side of (41):

$$
\begin{aligned}
&\nabla\cdot\Big\{\frac{1}{|\nabla u|^3}\nabla u\Big[\nabla u^T\nabla(\kappa|\nabla u|)\Big]\Big\}\\
=\ &\Big(\frac{\Delta u}{|\nabla u|^3}-\frac{3}{|\nabla u|^5}\alpha\Big)(\nabla u\cdot\mathbf{v})\\
=\ &\Big(\frac{\kappa\Delta u}{|\nabla u|^4}-\frac{(\Delta u)^2}{|\nabla u|^5}\Big)\alpha+\Big(\frac{3\Delta u}{|\nabla u|^7}-\frac{3\kappa}{|\nabla u|^6}\Big)\alpha^2-\frac{\Delta u}{|\nabla u|^5}\beta+\frac{3}{|\nabla u|^7}\alpha\beta
\end{aligned}
$$

Putting all three parts together, we have the expansion of $\nabla\cdot\mathbf{V}$ as

$$
\begin{aligned}
\nabla\cdot\mathbf{V} &= \kappa+b\kappa^3-\frac{2b\kappa}{|\nabla u|^4}(\alpha\Delta u+\beta)+2b\Big\{\Big(\frac{\Delta u}{|\nabla u|^5}+\frac{\kappa}{|\nabla u|^4}\Big)\beta+\frac{1}{|\nabla u|^5}\gamma\Big\}\\
&\quad+2b\Big\{\Big(\frac{\kappa\Delta u}{|\nabla u|^4}-\frac{(\Delta u)^2}{|\nabla u|^5}\Big)\alpha+\Big(\frac{3\Delta u}{|\nabla u|^7}-\frac{3\kappa}{|\nabla u|^6}\Big)\alpha^2-\frac{\Delta u}{|\nabla u|^5}\beta+\frac{3}{|\nabla u|^7}\alpha\beta\Big\}\\
&= \kappa+b\kappa^3-\frac{2b(\Delta u)^2}{|\nabla u|^5}\alpha+6b\Big(\frac{\Delta u}{|\nabla u|^7}-\frac{\kappa}{|\nabla u|^6}\Big)\alpha^2+\frac{6b}{|\nabla u|^7}\alpha\beta+\frac{2b}{|\nabla u|^5}\gamma.
\end{aligned}
$$

References

L. Ambrosio and S. Masnou. A direct variational approach to a problem arising in image reconstruction. *Interface and Free Boundaries*, 5:63–81, 2003.

L. Ambrosio, N. Fusco, and D. Pallara. *Functions of Bounded Variation and Free Discontinuity Problems*. Oxford University Press, 2000.

A. Asuncion and D.J. Newman. *UCI Machine Learning Repository*. 2013. URL `http://archive.ics.uci.edu/ml/`.

G. Aubert and P. Kornprobst. *Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations*. Springer-Verlag, 2nd edition, 2006.

E. Bae, J. Shi, and X.C. Tai. Graph cuts for curvature based image denoising. *IEEE Transaction on Image Processing*, 20(5):1199–1210, 2011.

P.L. Bartlett and M. Traskin. Adaboost is consistent. *Journal of Machine Learning Research*, 8:2347–2368, 2007.

P.L. Bartlett, M.I. Jordan, and J.D. McAuliffe. Convexity, classification, and risk bounds. *Journal of American Statistical Association*, 101(473):138–156, March 2006.

M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7(48):2399–2434, 2006.

G. Biau, L. Devroye, and G. Lugosi. Consistency of random forests and other averaging classifiers. *Journal of Machine Learning Research*, 9:2015–2033, 2008.

C.M. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag, 2006.

S. Boucheron, O. Bousquet, and G. Lugosi. Theory of classification: A survey of recent advances. *ESAIM: Probability and Statistics*, 9:323–375, 2005.

S. Boucheron, G. Lugosi, and P. Massart. *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford University Press, 2013.

O. Bousquet and A. Elisseeff. Stability and generalization. *Journal of Machine Learning Research*, 2:499–526, 2002.

O. Bousquet, S. Boucheron, and G. Lugosi. Introduction to statistical learning theory. In *Advanced Lectures in Machine Learning*. Springer, 2004.

K. Bredies, T. Bock, and B. Wirth. A convex, lower semi-continuous approximation of euler's elastica energy. Preprint, 2013.

L. Breiman. Random forest. *Machine Learning*, 45(1):5–32, 2001.

R. Caruana and A. Niculescu-Mizil. An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 161–168, Pittsburgh, Pennsylvania, 2006.

T.F. Chan and J. Shen. Nontexture inpainting by curvature driven diffusions (CDD). *Journal of Visual Communication and Image Representation*, 12:436–449, 2001.

T.F. Chan and J. Shen. *Image Processing and Analysis: Variational, PDE, Wavelet, and Stochastic Methods*. SIAM, 2005.

T.F. Chan, S.H. Kang, and J. Shen. Euler's elastica and curvature-based inpaintings. *SIAM Journal on Applied Mathematics*, 63:564–592, 2002.

C.C. Chang and C.J. Lin. Libsvm – a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:1–27, 2011. URL `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`.

J.B. Conway. *A Course in Functional Analysis*. Springer-Verlag, 2nd edition, 1990.

D. Cossock and T. Zhang. Statistical analysis of Bayes optimal subset ranking. *IEEE Transaction on Information Theory*, 54(11):5140–5154, 2008.

N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2000.

M. do Carmo. *Differential Geometry of Curves and Surfaces*. Prentice-Hall, 1976.

Y. Duan, Y. Wang, and J. Hahn. A fast augmented lagrangian algorithm for euler's elastica models. *Numerical Mathematics: Theory, Methods and Applications*, 6(1):47–71, 2013.

J.C. Duchi, L.W. Mackey, and M.I. Jordan. On the consistency of ranking algorithm. In *Proceedings of the 27th International Conference on Machine Learning*, pages 327–334, Haifa, Israel, 2010.

M. Fernández-Delgado, E. Cernadas, and S. Barro. Do we need hundreds of classifiers to solve real world classification problems? *Journal of Machine Learning Research*, 15: 3133–3181, 2014.

M.S. Floater and A. Iske. Multistep scattered data interpolation using compactly supported radial basis functions. *Journal of Computational and Applied Mathematics*, 73(1–2):65–78, 1996.

C.G. Fraser. Mathematical technique and physical conception in Euler's investigation of the elastica. *Centaurus*, 34(3):211–246, 1991.

W. Gao and Z.H. Zhou. On the consistency of multi-label learning. *Artificial Intelligence*, 199–200:22–44, 2013.

E. Giusti. *Minimal Surfaces and Functions of Bounded Variation*. Birkhäuser, Boston, 1994.

T. Glasmachers. Universal consistency of multi-class support vector classication. In *Proceedings of the 24th Annual Conference on Neural Information Processing Systems*, Vancouver, Canada, 2010.

B.I. Golubov and A.G. Vitushkin. Variation of a function. In M. Hazewinkel, editor, *Encyclopedia of Mathematics*. Springer, 2001. URL `http://www.encyclopediaofmath.org/index.php/Function_of_bounded_variation`, updated in 2013.

J. Hahn, G.J. Chung, Y. Wang, and X.C. Tai. Fast algorithms for p-elastica energy with the application to image inpaiting and curve reconstruction. In *Proceedings of International Conference on Scale Space and Variational Methods in Computer Vision*, pages 169–182. Springer, 2011.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer-Verlag, 2009.

C.-W. Hsu and C.-J. Lin. A comparison of methods for multi-class support vector machines. *IEEE Transactions on Neural Networks*, 13(2):415–425, 2002.

C.-W. Hsu, C.-C. Chang, and C.-J. Lin. *A practical guide to support vector classification*. 2007. URL `http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf`.

X. Huang, L. Shi, and J.A.K. Suykens. Ramp loss linear programming support vector machine. *Journal of Machine Learing Research*, 15(6):2185–2211, 2014.

J.K. Hunter. Chapter 7: L^p spaces. *Course Notes of Measure Theory*, 2011. URL `http://www.math.ucdavis.edu/~hunter/m206/ch6_measure_notes.pdf`.

G. Kanizsa. *Organization in Vision*. Praeger, New York, 1979.

N. Komodakis and N. Paragios. Beyond pairwise energies: efficient optimization for higher-order MRFs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2985–2992, 2009.

S. Kutin and P. Niyogi. Almost everywhere algorithmic stability and generalization error. In *Proceedings of the Eighteenth Conference Conference on Uncertainty in Artificial Intelligence*, pages 275–282, Alberta,Canada, 2002.

J.M. Lee. *Riemannian Manifolds: An Introduction to Curvature*. Springer-Verlag, 1997.

G.P. Leonardi and S. Masnou. Locality of the mean curvature of rectifiable varifolds. *Advances in Calculus of Variations*, 2(1):17–42, 2009.

R. Levien. The elastica: a mathematical history. Technical report, EECS Department, University of California, Berkeley, 2008.

S. Masnou and J.-M. Morel. Level lines based disocclusion. In *Proceedings of the 5th IEEE International Conference on Image Processing*, pages 259–263, Chicago, Illinois, October 1998.

M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of Machine Learning*. MIT Press, 2012.

J.M. Morel and S. Solimini. Variational methods in image segmentation. In *Progress in Nonlinear Differential Equations and Their Applications*. Birkhäuser, Boston, 1995.

D. Mumford. Elastica and computer vision. In C.L. Bajaj, editor, *Algebraic Geometry and Its Applications*, pages 491–506. Springer-Verlag, New York, 1994.

K.P. Murphy. *Machine Learning: a Probabilistic Perspective*. MIT Press, 2012.

B. Nadler, N. Srebro, and X. Zhou. Semi-supervised learning with the graph laplacian: The limit of infinite unlabelled data. In *Proceedings of the 24th Annual Conference on Neural Information Processing Systems*, pages 1330–1338, Vancouver, B.C., Canada, 2009.

M. Nandan, P.P. Khargonekar, and S.S. Talathi. Fast SVM training using approximate extreme points. *Journal of Machine Learning Research*, 15(1):59–98, 2014.

F. Nie, Y. Huang, and H. Huang. Linear time solver for primal SVM. In *Proceedings of The 31st International Conference on Machine Learning*, pages 505–513, Beijing, 2014.

S. Osher and J.A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79(1):12–49, 1988.

T. Poggio, S. Rifkin, S. Mukherjee, and P. Niyogi. General conditions for predictivity in learning theory. *Nature*, 428:419–422, 2004.

R. M. Rifkin. *Everything Old Is New Again : A Fresh Look at Historical Approaches in Machine Learning*. PhD thesis, MIT, 2002.

L.I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60:259–268, 1992.

R.E. Schapire and Y. Freund. *Boosting: Foundations and Algorithms*. MIT Press, 2012.

B. Schölkopf and A. Smola. *Learning with Kernels*. MIT Press, 2002.

M. R. Spiegel and S. Lipschutz. *Vector Analysis*. McGraw-Hill, 2nd edition, 2009.

M. Spivak. *A Comprehensive Introduction to Differential Geometry*, volume 3–4. Publish or Perish Press, 3rd edition, 1999.

I. Steinwart. Consistency of support vector machines and other regularized kernel classifiers. *IEEE Transactions on Information Theory*, 51(1):128–142, 2005.

X.C. Tai, J. Hahn, and G.J. Chung. A fast algorithm for euler's elastica model using augmented lagrangian method. *SIAM Journal on Imaging Sciences*, 4(1):313–344, 2011.

A. Tewari and P.L. Bartlett. On the consistency of multiclass classification methods. *Journal of Machine Learning Research*, 8:1007–1025, 2007.

B. van Brunt. *The Calculus of Variations*. Springer-Verlag, 2004.

V.N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, 1998.

K. R. Varshney and A. S. Willsky. Classification using geometric level sets. *Journal of Machine Learning Research*, 11(2):491–516, 2010.

U. von Luxburg and B. Schölkopf. Statistical learning theory: Models, concepts, and results. Technical report, arXiv:0810.4752, 2008.

J. Wang, P. Wonka, and J. Ye. Scaling SVM and least absolute deviations via exact data reduction. In *Proceedings of The 31st International Conference on Machine Learning*, pages 523–531, Beijing, 2014.

H. Wendland. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Advances in Computational Mathematics*, 4(1):389–396, 1995.

F. Xia, T.Y. Liu, J. Wang, W. Zhang, and H. Li. Listwise approach to learning to rank: Theory and algorithm. In *Proceedings of the 25th International Conference on Machine Learning*, pages 1192–1199, Helsinki, Finland, 2008.

K. Yosida. *Functional Analysis*. Springer-Verlag, 6th edition, 1999.

A. Zakai and Y. Ritov. Consistency and localizability. *Journal of Machine Learning Research*, 10:827–856, 2009.

T. Zhang. Statistical behavior and consistency of classification methods based on convex risk minimization. *Annals of Statistics*, 32(1):56–85, 2004a.

T. Zhang. Statistical analysis of some multi-category large margin classification methods. *Journal of Machine Learning Research*, 5:1225–1251, 2004b.

D. Zhou and B. Schölkopf. Regularization on discrete spaces. In *Proceedings of the 27th DAGM Symposium Symposium on Pattern Recognition*, pages 361–368, Springer, Berlin, 2005.

Journal of Machine Learning Research 16 (2015) 3687-3720 Submitted 4/14; Revised 2/15; Published 12/15

Learning to Identify Concise Regular Expressions that Describe Email Campaigns

Paul Prasse PRASSE@CS.UNI-POTSDAM.DE
University of Potsdam, Department of Computer Science
August-Bebel-Strasse 89, 14482 Potsdam, Germany

Christoph Sawade CHRISTOPH@SOUNDCLOUD.COM
SoundCloud Ltd.
Rheinsberger Str. 76/77, 10115 Berlin, Germany

Niels Landwehr LANDWEHR@CS.UNI-POTSDAM.DE
Tobias Scheffer SCHEFFER@CS.UNI-POTSDAM.DE
University of Potsdam, Department of Computer Science
August-Bebel-Strasse 89, 14482 Potsdam, Germany

Editor: Ivan Titov

Abstract

This paper addresses the problem of inferring a regular expression from a given set of strings that resembles, as closely as possible, the regular expression that a human expert would have written to identify the language. This is motivated by our goal of automating the task of postmasters who use regular expressions to describe and blacklist email spam campaigns. Training data contains batches of messages and corresponding regular expressions that an expert postmaster feels confident to blacklist. We model this task as a two-stage learning problem with structured output spaces and appropriate loss functions. We derive decoders and the resulting optimization problems which can be solved using standard cutting plane methods. We report on a case study conducted with an email service provider.

Keywords: applications of machine learning, learning with structured output spaces, supervised learning, regular expressions, email campaigns

1. Introduction

The problem setting introduced in this paper is motivated by the intuition of *automatically reverse engineering* email spam campaigns. Email-spam generation tools allow users to implement mailing campaigns by specifying simple grammars that serve as message templates. A grammar is disseminated to nodes of a bot net; the nodes create messages by instantiating the grammar at random. Email service providers can easily sample elements of new mailing campaigns by collecting messages in spam traps or by tapping into known bot nets. When messages from multiple campaigns are collected in a joint spam trap, clustering tools can separate the campaigns reliably (Haider and Scheffer, 2009). However, probabilistic cluster descriptions that use a bag-of-words representation incur the risk of false positives, and it is difficult for a human to decide whether they in fact characterize the correct set of messages.

Typically, mailing campaigns are quite specific. A specific, comprehensible regular expression written by an expert postmaster can be used to blacklist the bulk of emails of that campaign at virtually no risk of covering any other messages. This, however, requires the continuous involvement of a human postmaster.

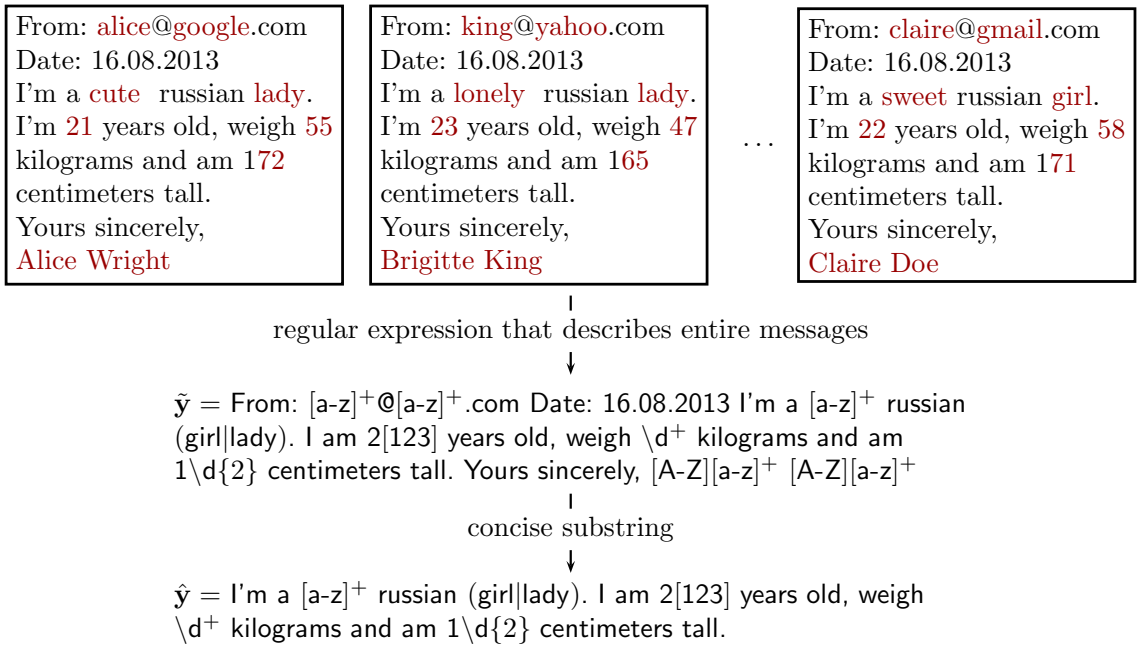


Figure 1: Elements of a message spam campaign, a regular expression that describes the entirety of the messages, and a concise regular expression that describes a characteristic substring of the messages.

Regular expressions are a standard tool for specifying simple grammars. Widely available tools match strings against regular expressions efficiently and can be used conveniently from scripting languages. A regular expression can be translated into a deterministic finite automaton that accepts the language and has an execution time linear in the length of the input string.

Language identification has a rich history in the algorithmic learning theory community, see Section 6 for a brief review. Our problem setting reflects the process that we seek to automate; it differs from the classical problem of language identification in the learner's exact goal, and in the available training data. Batches of strings and corresponding regular expressions are observable in the training data. These regular expressions have been written by postmasters to blacklist mailing campaigns. The learner's goal is to produce a predictive model that maps batches of strings to regular expressions that resemble, as closely as possible, the regular expressions which the postmaster would have written and feels confident to blacklist. As an illustration of this problem, Figure 1 shows three messages of a mailing campaign, a regular expression that describes the entirety of the messages, and

a more concise regular expression that characterizes a characteristic substring, and that a postmaster has selected to blacklist the corresponding email campaign.

This paper extends a conference publication (Prasse et al., 2012) that addresses this problem setting with linear models and structured output spaces. In the decoding step, a set of strings is given and the space of all regular expressions has to be searched for an element that maximizes the decision function. Since this space is very large and difficult to search, the approach of Prasse et al. (2012) is constrained to finding specializations of an approximate maximal alignment of all strings. The maximal alignment is a regular expression that contains all character sequences which occur in each of the strings, and uses wildcards wherever there are differences between the strings.

The maximal alignment is extremely specific. By constraining the output to specializations of the alignment, the method keeps the risk that any message which is not part of the same campaign is accidentally matched at a minimum. However, since all specializations of this alignment describe the entire length of the strings, the method produces regular expressions that tend to be much longer than the more concise expressions that postmasters prefer. Also, as a consequence of their greater length, the finite state automata which correspond to these expressions tend to have more states, which limits the number of regular expressions that can be matched in parallel against incoming new messages. This paper therefore extends the method by including a mechanism which learns to select expressions that describe only the most characteristic part of the mailing campaign, using regular expressions written by an expert postmaster as training data.

The rest of this paper is structured as follows. Section 2 reviews regular expressions before Section 3 states the problem setting. Section 4 introduces the feature representations and derives the decoders and the optimization problems. In Section 5, we discuss our findings from a case study with an email service. Section 6 discusses related work and Section 7 concludes.

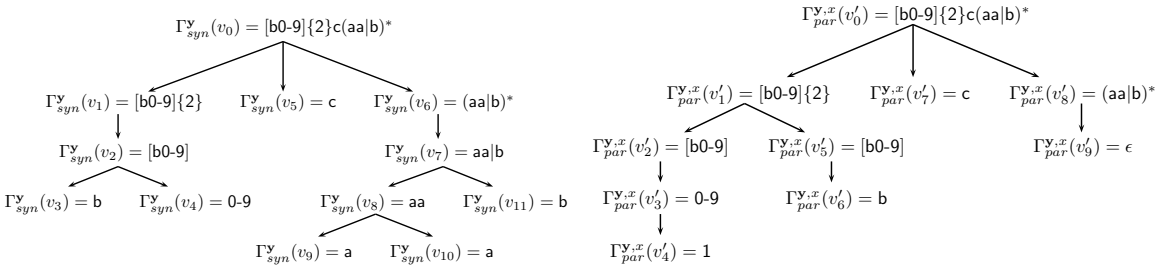
2. Regular Expressions

Before we formulate the problem setting, let us briefly revisit the syntax and semantics of regular expressions. Regular expressions are a popular syntactic convention for the definition of regular languages. Syntactically, a regular expression $\mathbf{y} \in \mathcal{Y}_\Sigma$ is either a character from an alphabet Σ, or it is an expression in which an operator is applied to one or several argument expressions. Basic operators are the concatenation (*e.g.,* "abc"), disjunction (*e.g.,* "a|b"), and the *Kleene* star ("*"), written in postfix notation ("(abc)*"), that accepts any number of repetitions of its preceding argument expression. Parentheses define the syntactic structure of the expression. For better readability, several shorthands are used, which can be defined in terms of the basic operators. For instance, the *any character* symbol (".") abbreviates the disjunction of all characters in Σ, square brackets accept the disjunction of all characters (*e.g.,* "[abc]") or ranges (*e.g.,* "[a-z0-9]") that are included. For instance, the regular expression [a-z0-9] accepts all lower-case letters and digits. The postfix operator "+" accepts an arbitrary, positive number of reiterations of the preceding expression, while "$\{l, u\}$" accepts between l and u reiterations, where $l \leq u$. We include a set of popular macros—for instance "\d" for *any digit* or the macro "\e" for all

characters, which can occur in a URL. A formal definition of the set of regular expressions can be found in Definition 3 in the appendix.

The set of all regular expressions can be described by a context-free language. The syntactic structure of a regular expression $\mathbf{y}$ is typically represented by its *syntax tree* $T^{\mathbf{y}}_{syn} = (V^{\mathbf{y}}_{syn}, E^{\mathbf{y}}_{syn}, \Gamma^{\mathbf{y}}_{syn}, \leq^{\mathbf{y}}_{syn})$. Definition 4 in the appendix assigns one such tree to each regular expression. Each node $v \in V^{\mathbf{y}}_{syn}$ of this syntax tree is tagged by a labeling function $\Gamma^{\mathbf{y}}_{syn} : V^{\mathbf{y}}_{syn} \rightarrow \mathcal{Y}_\Sigma$ with a subexpression $\Gamma^{\mathbf{y}}_{syn}(v) = \mathbf{y}_j$. The edges $(v, v') \in E^{\mathbf{y}}_{syn}$ indicate that node v' represents an argument expression of v. Relation $\leq^{\mathbf{y}}_{syn} \subseteq V^{\mathbf{y}}_{syn} \times V^{\mathbf{y}}_{syn}$ defines an ordering on the nodes and identifies the root node. Note that the root node is labeled with the entire regular expression $\mathbf{y}$.

A regular expression $\mathbf{y}$ defines a regular language $L(\mathbf{y})$. Given the regular expression, a deterministic finite state machine can decide whether a string x is in $L(\mathbf{y})$ in time linear in $|x|$ (Dubé and Feeley, 2000). The trace of verification is typically represented as a *parse tree* $T^{\mathbf{y},x}_{par} = (V^{\mathbf{y},x}_{par}, E^{\mathbf{y},x}_{par}, \Gamma^{\mathbf{y},x}_{par}, \leq^{\mathbf{y},x}_{par})$, describing how the string x can be derived from the regular expression $\mathbf{y}$. At least one parse tree exists if and only if the string is an element of the language $L(\mathbf{y})$; in this case, $\mathbf{y}$ is said to generate x. Multiple parse trees can exist for one regular expression $\mathbf{y}$ and a string x. Nodes $v \in V^{\mathbf{y}}_{syn}$ of the syntax tree generate the nodes of the parse tree $v' \in V^{\mathbf{y},x}_{par}$, where nodes of the syntax tree may spawn none (alternatives which are not used to generate a string), one, or several ("loopy" syntactic elements such as "*" or "+") nodes in the parse tree. In analogy to the syntax trees, the labeling function $\Gamma^{\mathbf{y},x}_{par} : V^{\mathbf{y},x}_{par} \rightarrow \mathcal{Y}_\Sigma$ assigns a subexpression to each node, and the relation $\leq^{\mathbf{y},x}_{par} \subseteq V^{\mathbf{y},x}_{par} \times V^{\mathbf{y},x}_{par}$ defines the ordering of sibling nodes. The set of all parse trees for a regular expression $\mathbf{y}$ and a string x is denoted by $\mathcal{T}^{\mathbf{y},x}_{par}$. When multiple parse trees exist for a regular expression and a string, a canonical parse tree can be selected by choosing the *left-most parse*. Standard tools for regular expressions typically follow this convention and generate the left-most parse tree. Definition 5 in the appendix gives a formal definition.



(a) Syntax tree $T^{\mathbf{y}}_{syn}$ (b) Parse tree $T^{\mathbf{y},x}_{par}$

Figure 2: Syntax tree (a) and a parse tree (b) for the regular expression $\mathbf{y} =$ [b0-9]{2}c(aa|b)* and the string $x =$ 1bc.

Leaf nodes of a parse tree $T^{\mathbf{y},x}_{par}$ are labeled with elements of $\Sigma \cup \{\epsilon\}$, where ϵ denotes the empty symbol; reading them from left to right gives the generated string x. Non-terminal nodes correspond to subexpressions $\mathbf{y}_j$ of $\mathbf{y}$ which generate substrings of x. To compare different regular expressions with respect to a given string x, we define the set $T^{\mathbf{y},x}_{par\,|i}$ of

labels of nodes which are visited on the path from the root to the the i-th character of x in the parse tree $T^{\mathbf{y},x}_{par}$.

Figure 2 (left) shows an example of a syntax tree $T^{\mathbf{y}}_{syn}$ for the regular expression $\mathbf{y} =$ [b0-9]{2}c(aa|b)*. One corresponding parse tree $T^{\mathbf{y},x}_{par}$ for the string $x =$ 1bc is illustrated in Figure 2 (right). The set $T^{\mathbf{y},x}_{par\,|2}$ contains nodes v'_0, v'_1, v'_5, and v'_6.

Finally, we introduce the concept of a matching list. When a regular expression $\mathbf{y}$ generates a set $\mathbf{x}$ of strings, and $v \in V^{\mathbf{y}}_{syn}$ is an arbitrary node of the syntax tree of $\mathbf{y}$, then the matching list $M^{\mathbf{y},\mathbf{x}}(v)$ characterizes which substrings of the strings in $\mathbf{x}$ are generated by the node v of the syntax tree, and thus generated by the subexpression $\Gamma^{\mathbf{y}}_{syn}(v)$. A node v of the syntax tree generates a substring x' of $x \in \mathbf{x}$, if v generates a node v' in the parse tree $T^{\mathbf{y},x}_{par}$ of x, and there is a path from v' in that parse tree to every character in the substring x'. In the above example, for the set of strings $\mathbf{x} = \{$12c, b4ca$\}$, the matching list for node v_1 that represents subexpression $\Gamma^{\mathbf{y}}_{syn}(v_1) =$ [b0-9]{2} is $M^{\mathbf{y},\mathbf{x}}(v_2) = \{$12, b4$\}$. Definition 5 in the appendix introduces matching lists more formally.

3. Problem Setting

Having established the syntax and semantics of regular expressions, we now define our problem setting. An unknown distribution $p(\mathbf{x}, \mathbf{y})$ generates regular expressions $\mathbf{y} \in \mathcal{Y}_\Sigma$ from the alphabet Σ and batches $\mathbf{x}$ of strings $x \in \mathbf{x}$ that are elements of the language $L(\mathbf{y})$. In our motivating application, the strings x are messages that belong to one particular mailing campaign and have been sampled from a bot net, and the $\mathbf{y}$ are regular expressions which an expert postmaster believes to identify the campaign template, and feels highly confident to blacklist.

A $\mathbf{w}$-parameterized predictive model $f_\mathbf{w} : \mathbf{x} \times \hat{\mathbf{y}} \mapsto \mathbb{R}$ maps a batch of strings and a regular expression $\hat{\mathbf{y}}$ to a value of the decision function. We refer to the process of inferring the $\hat{\mathbf{y}}$ that attains the highest score $f_\mathbf{w}(\mathbf{x}, \hat{\mathbf{y}})$ for a given batch of strings $\mathbf{x}$ as *decoding*; in this step, a decision function is maximized over $\hat{\mathbf{y}}$ which generally involves a search over the space of all regular expressions.

A loss function $\Delta(\mathbf{y}, \hat{\mathbf{y}}, \mathbf{x})$ quantifies the difference between the true and predicted expressions. While it would, in principle, be possible to use the zero-one loss $\Delta_{0/1}(\mathbf{y}, \hat{\mathbf{y}}, \mathbf{x}) = [\![\mathbf{y} = \hat{\mathbf{y}}]\!]$, this loss function would treat nearly-identical expressions and very dissimilar expressions alike. We will later engineer a loss function whose gradient will guide the learner towards expressions $\hat{\mathbf{y}}$ that are more similar to the correct expression $\mathbf{y}$.

In the learning step, the ultimate goal is to identify parameters that minimize the risk—the expected loss—under the unknown distribution $p(\mathbf{x}, \mathbf{y})$:

$$R[f_\mathbf{w}] = \iint \Delta\left(\mathbf{y}, \arg\max_{\hat{\mathbf{y}} \in \mathcal{Y}_\Sigma} f_\mathbf{w}(\mathbf{x}, \hat{\mathbf{y}}), \mathbf{x}\right) p(\mathbf{x}, \mathbf{y}) \mathrm{d}\mathbf{x}\, \mathrm{d}\mathbf{y}.$$

The underlying distribution $p(\mathbf{x}, \mathbf{y})$ is not known, and therefore this goal is unattainable. We resort to training data $D = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^m$ that consists of pairs of batches $\mathbf{x}_i$ and corresponding regular expressions $\mathbf{y}_i$, drawn according to $p(\mathbf{x}, \mathbf{y})$. In order to obtain a convex optimization problem that can be evaluated using the training data, we approximate the risk by the hinged upper bound of its maximum-likelihood estimate, following

the margin-rescaling approach (Tsochantaridis et al., 2005), with added regularization term $\Omega(\mathbf{w})$:

$$\hat{R}[f_{\mathbf{w}}] = \frac{1}{m}\sum_{i=1}^{m} \max_{\bar{y}} \left\{ f_{\mathbf{w}}(\mathbf{x}_i, \bar{\mathbf{y}}) - f_{\mathbf{w}}(\mathbf{x}_i, \mathbf{y}_i) + \Delta(\mathbf{y}, \bar{\mathbf{y}}, \mathbf{x}_i), 0 \right\} + \Omega(\mathbf{w}). \tag{1}$$

This problem setting differs fundamentally from traditional language identification settings. In our setting, the actual identification of a language from example strings takes place in the decoding step. In this step, the decoder searches the space of regular expressions. But instead of retrieving an expression that generates all strings in $\mathbf{x}$, it searches for an expression that maximizes the value of a $\mathbf{w}$-parameterized decision function that receives the strings and the candidate expression as arguments. In a separate learning step, the parameters $\mathbf{w}$ are optimized using batches of strings and corresponding regular expressions. The training process has to optimize the model parameters $\mathbf{w}$ such that the expected deviation between the decoder's output and a regular expression written by a human postmaster is minimized. Training data of this form, and an optimization criterion that measures the expected discrepancy between the conjectured regular expressions and regular expressions written by a human labeler, are not part of traditional language identification settings.

4. Identifying Regular Expressions

This section details our approach to identifying regular expressions based on generalized linear models and structured output spaces.

4.1 Problem Decomposition

Without any approximations, the decoding problem—the problem of identifying the regular expression $\mathbf{y}$ that maximizes the parametric decision function—is insurmountable. For any string, an exponential number of matching regular expressions of up to the same length can be constructed by substituting constant symbols for wildcards. In addition, constant symbols can be replaced by disjunctions and "loopy" syntactic elements can be added to create infinitely many longer regular expressions that also match the original string. Because the space of regular expressions is discrete, it also does not lend itself well to approaches based on gradient descent.

We decompose the problem into two more strongly constrained learning problems. We decompose the parameters $\mathbf{w} = (\mathbf{u}\ \mathbf{v})^{\mathsf{T}}$ and the loss function $\Delta = \Delta_{\mathbf{u}} + \Delta_{\mathbf{v}}$ into parts that are minimized sequentially. In the first step, $\mathbf{u}$-parameterized model $f_{\mathbf{u}}$ produces a regular expression $\tilde{\mathbf{y}}$ that is constrained to being a specialization of the maximal alignment of the strings in $\mathbf{x}$. Specializations of maximal alignments of the strings in $\mathbf{x}$ tend to be long regular expressions that characterize the entirety of the strings in $\mathbf{x}$. In a second step, $\mathbf{v}$-parameterized model $f_{\mathbf{v}}$ therefore produces a concise substring $\hat{\mathbf{y}}$ of $\tilde{\mathbf{y}}$.

Definition 1 (Alignment, Maximal Alignment) *The set of* alignments $A_{\mathbf{x}}$ *of a batch of strings* $\mathbf{x}$ *contains all concatenations in which strings from* Σ^+ *and the wildcard symbol "(.*)" alternate, and that generates all elements of* $\mathbf{x}$*.The set of* maximal alignments $A^*_{\mathbf{x}} \subseteq A_{\mathbf{x}}$ *contains all alignments of the strings in* $\mathbf{x}$ *which share the property that no other alignment in* $A_{\mathbf{x}}$ *has more constant symbols.*

A specialization of an alignment is a string that has been derived from an alignment by replacing one or several wildcard symbols by another regular expression. Figure 3 illustrates the process of generating a maximal alignment, and the subsequent step of specializing it.

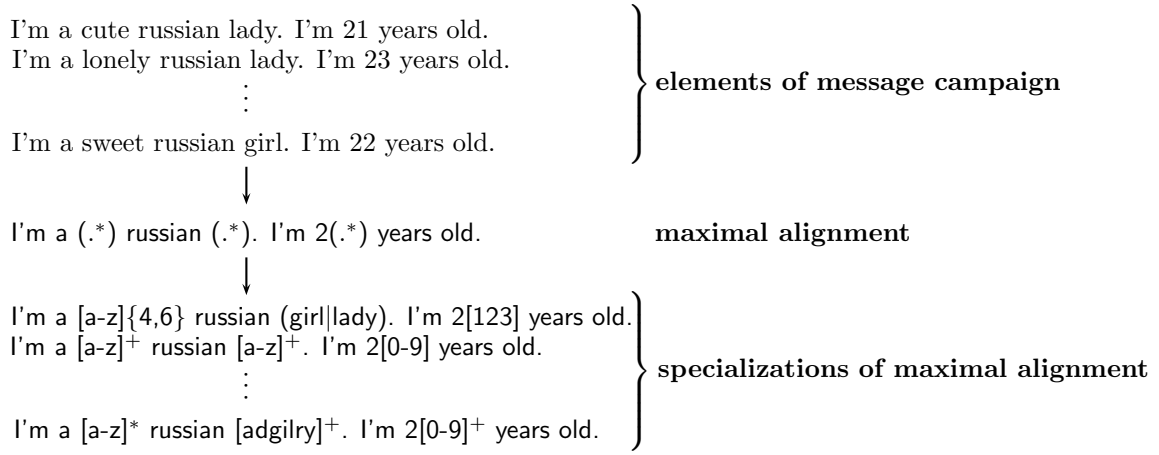


Figure 3: Examples of regular expressions, which are specializations of a maximal alignment of strings.

The loss function for this step should measure the semantic and syntactic deviation between the conjecture $\tilde{\mathbf{y}}$ and the manually written $\mathbf{y}$ for batch $\mathbf{x}$. We define a loss function $\Delta_{\mathbf{u}}(\mathbf{y}, \tilde{\mathbf{y}}, \mathbf{x})$ that compares the set of parse trees in $\mathcal{T}_{par}^{\mathbf{y},x}$, for each string $x \in \mathbf{x}$ to the most similar tree in $\mathcal{T}_{par}^{\tilde{\mathbf{y}},x}$; if no such parse tree exists, the summand is defined as $\frac{1}{|\mathbf{x}|}$ (Equation 2). Similarly to a loss function for hierarchical classification (Cesa-Bianchi et al., 2006), the difference of two parse trees for a given string x is quantified by a comparison of the paths that lead to the characters of the string. Two paths are compared by means of the intersection of their nodes (Equation 3). This loss is bounded between zero and one; it is zero if and only if the two regular expressions $\tilde{\mathbf{y}}$ and $\mathbf{y}$ are equal:

$$\Delta_{\mathbf{u}}(\mathbf{y}, \tilde{\mathbf{y}}, \mathbf{x}) = \frac{1}{|\mathbf{x}|} \sum_{x \in \mathbf{x}} \begin{cases} \Delta_{\text{tree}}(\mathbf{y}, \tilde{\mathbf{y}}, x) & \text{if } x \in L(\tilde{\mathbf{y}}) \\ 1 & \text{otherwise} \end{cases} \tag{2}$$

$$\text{with } \Delta_{\text{tree}}(\mathbf{y}, \tilde{\mathbf{y}}, x) = 1 - \frac{1}{|\mathcal{T}_{par}^{\mathbf{y},x}|} \sum_{t \in \mathcal{T}_{par}^{\mathbf{y},x}} \max_{\tilde{t} \in \mathcal{T}_{par}^{\tilde{\mathbf{y}},x}} \frac{1}{|x|} \sum_{j=1}^{|x|} \frac{|t_{|j} \cap \tilde{t}_{|j}|}{\max\{|t_{|j}|, |\tilde{t}_{|j}|\}} \tag{3}$$

Figure 4 illustrates how the tree loss is calculated for a single string: for each symbol, the corresponding paths of the syntax trees spawned by $\mathbf{y}$ and $\tilde{\mathbf{y}}$ are compared. Each pair of corresponding paths incurs a loss according to the proportion of nodes that are labeled with differing subexpressions.

Because the regular expression created in this step is a specialization of a *maximal* alignment, it is not generally concise. In the second step, $\mathbf{v}$-parameterized model $f_{\mathbf{v}}$ produces

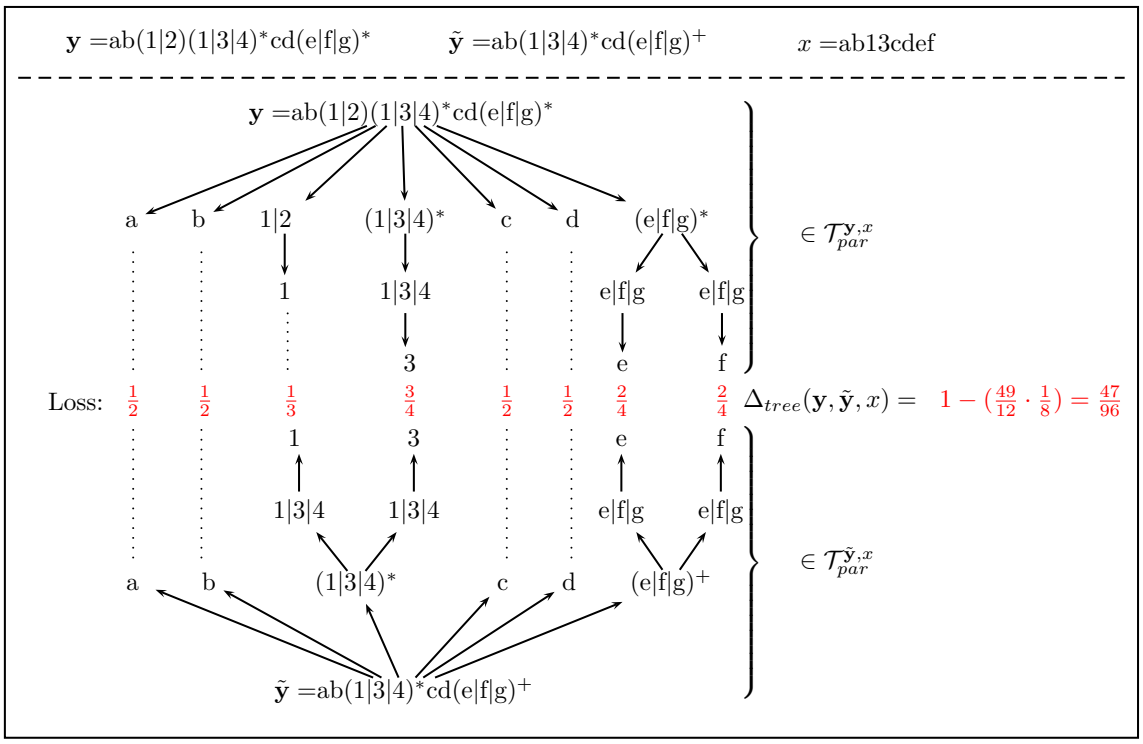


Figure 4: Calculation of the tree loss $\Delta_{tree}(\mathbf{y}, \tilde{\mathbf{y}}, x)$ for a given string x and two regular expressions $\mathbf{y}$ and $\tilde{\mathbf{y}}$.

a regular expression $\hat{\mathbf{y}} \in \mathcal{Y}_\Sigma$ that is a subexpression of $\tilde{\mathbf{y}}$; that is, $\tilde{\mathbf{y}} = \mathbf{y}_{\text{pre}}\hat{\mathbf{y}}\mathbf{y}_{\text{suf}}$ with $\mathbf{y}_{\text{pre}}, \mathbf{y}_{\text{suf}} \in \mathcal{Y}_\Sigma$. Loss function $\Delta_{\mathbf{v}}(\mathbf{y}, \hat{\mathbf{y}})$ is based on the length of the longest common substring $\text{lcs}(\mathbf{y}, \hat{\mathbf{y}})$ of $\mathbf{y}$ and $\hat{\mathbf{y}}$. The loss—defined in Equation 4—is zero, if the longest common substring of $\mathbf{y}$ and $\hat{\mathbf{y}}$ is equal to both $\mathbf{y}$ and $\hat{\mathbf{y}}$. In this case, $\mathbf{y} = \hat{\mathbf{y}}$. Otherwise, it increases as the longest common substring of $\mathbf{y}$ and $\hat{\mathbf{y}}$ decreases:

$$\Delta_{\mathbf{v}}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{2}\left[\left(\frac{|\mathbf{y}| - |\text{lcs}(\mathbf{y}, \hat{\mathbf{y}})|}{|\mathbf{y}|}\right) + \left(\frac{|\hat{\mathbf{y}}| - |\text{lcs}(\mathbf{y}, \hat{\mathbf{y}})|}{|\hat{\mathbf{y}}|}\right)\right]. \tag{4}$$

In the following subsections, we derive decoders and optimization problems for these two subproblems.

4.2 Learning to Generate Regular Expressions

We model $f_{\mathbf{u}}$ as a linear discriminant function $\mathbf{u}^\mathsf{T}\Psi_{\mathbf{u}}(\mathbf{x}, \mathbf{y})$ for a joint feature representation of the input $\mathbf{x}$ and output $\mathbf{y}$ (Tsochantaridis et al., 2005):

$$\tilde{\mathbf{y}} = \arg\max_{\mathbf{y} \in \mathcal{Y}_\Sigma} f_{\mathbf{u}}(\mathbf{x}, \mathbf{y}) = \arg\max_{\mathbf{y} \in \mathcal{Y}_\Sigma} \mathbf{u}^\mathsf{T}\Psi_{\mathbf{u}}(\mathbf{x}, \mathbf{y}).$$

4.2.1 Joint Feature Representation for Generating Regular Expressions

The joint feature representation $\Psi_{\mathbf{u}}(\mathbf{x}, \mathbf{y})$ captures structural properties of an expression $\mathbf{y}$ and joint properties of input batch $\mathbf{x}$ and regular expression $\mathbf{y}$.

It captures structural properties of a regular expression $\mathbf{y}$ by features that indicate a specific nesting of regular expression operators—for instance, whether a concatenation occurs within a disjunction. More formally, we first define a binary vector

$$\Lambda_{\mathbf{u}}(\mathbf{y}) = \begin{pmatrix} [\![\mathbf{y} = \mathbf{y}_1 \dots \mathbf{y}_k]\!] \\ [\![\mathbf{y} = \mathbf{y}_1 | \dots | \mathbf{y}_k]\!] \\ [\![\mathbf{y} = [\mathbf{y}_1 \dots \mathbf{y}_k]]\!] \\ [\![\mathbf{y} = \mathbf{y}_1^*]\!] \\ [\![\mathbf{y} = \mathbf{y}_1?]\!] \\ [\![\mathbf{y} = \mathbf{y}_1^+]\!] \\ [\![\mathbf{y} = \mathbf{y}_1\{l\}]\!] \\ [\![\mathbf{y} = \mathbf{y}_1\{l, u\}]\!] \\ [\![\mathbf{y} = r_1]\!] \\ \vdots \\ [\![\mathbf{y} = r_l]\!] \\ [\![\mathbf{y} \in \Sigma]\!] \\ [\![\mathbf{y} = \epsilon]\!] \end{pmatrix} \tag{5}$$

that encodes the top-level operator used in the regular expression $\mathbf{y}$, where $[\![\cdot]\!]$ is the indicator function of its Boolean argument. In Equation 5, $\mathbf{y}_1, \dots, \mathbf{y}_k \in \mathcal{Y}_\Sigma$ are regular expressions, $l, u \in \mathbb{N}$, and $\{r_1, \dots, r_l\}$ is a set of ranges and popular macros. For our application, we use the set $\{$0-9, a-f, a-z, A-F, A-Z, \S, \e, \d, "."$\}$ (see Table 6 in the appendix) because these are frequently used by postmasters.

For any two nodes v' and v'' in the syntax tree of $\mathbf{y}$ that are connected by an edge—indicating that $\mathbf{y}'' = \Gamma^{\mathbf{y}}_{syn}(v'')$ is an argument subexpression of $\mathbf{y}' = \Gamma^{\mathbf{y}}_{syn}(v')$—the tensor product $\Lambda_{\mathbf{u}}(\mathbf{y}') \otimes \Lambda_{\mathbf{u}}(\mathbf{y}'')$ defines a binary vector that encodes the specific nesting of operators at node v'. Feature vector $\Psi_{\mathbf{u}}(\mathbf{x}, \mathbf{y})$ will aggregate these vectors over all pairs of adjacent nodes in the syntax tree of $\mathbf{y}$.

Joint properties of an input batch $\mathbf{x}$ and a regular expression $\mathbf{y}$ are encoded in a similar way as follows. Recall that for any node v' in the syntax tree, $M^{\mathbf{y},\mathbf{x}}(v')$ denotes the set of substrings in $\mathbf{x}$ that are generated by the subexpression $\mathbf{y}' = \Gamma^{\mathbf{y}}_{syn}(v')$ that v' is labeled with. We define a vector $\Phi_{\mathbf{u}}(M^{\mathbf{y},\mathbf{x}}(v'))$ of attributes of this set. Any property may be accounted for; for our application, we include the average string length, the inclusion of the empty string, the proportion of capital letters, and many other attributes. The list of attributes used in our experiments is included in the appendix in Table 3. A joint encoding of properties of the subexpression $\mathbf{y}'$ and the set of substrings generated by $\mathbf{y}'$ is given by the tensor product $\Phi_{\mathbf{u}}(M^{\mathbf{y},\mathbf{x}}(v')) \otimes \Lambda_{\mathbf{u}}(\mathbf{y}')$.

The joint feature vector $\Psi_{\mathbf{u}}(\mathbf{x}, \mathbf{y})$ is obtained by aggregating operator-nesting information over all edges in the syntax tree, and joint properties of subexpressions $\mathbf{y}'$ and the set of substrings which they generate over all nodes in the syntax tree:

$$\Psi_{\mathbf{u}}(\mathbf{x}, \mathbf{y}) = \begin{pmatrix} \sum_{(v',v'') \in E^{\mathbf{y}}_{syn}} \Lambda_{\mathbf{u}}(\Gamma^{\mathbf{y}}_{syn}(v')) \otimes \Lambda_{\mathbf{u}}(\Gamma^{\mathbf{y}}_{syn}(v'')) \\ \sum_{v' \in V^{\mathbf{y}}_{syn}} \Phi_{\mathbf{u}}(M^{\mathbf{y},\mathbf{x}}(v')) \otimes \Lambda_{\mathbf{u}}(\Gamma^{\mathbf{y}}_{syn}(v')) \end{pmatrix}. \tag{6}$$

4.2.2 Decoding Specializations of the Maximal Alignment

At application time, the highest-scoring regular expression $\tilde{\mathbf{y}}$ according to model $f_{\mathbf{u}}$ has to be decoded. Model $f_{\mathbf{u}}$ is constrained to producing specializations of the maximal alignment; however, searching the space of all possible specializations of the maximal alignment is still not feasible. The following observation illustrates that $f_{\mathbf{u}}$ may not even have a maximum, because there may always be a longer expression that attains a higher score.

Observation 1 *Given a string* $\mathbf{a}$ *that contains at least one wildcard symbol "(.*)", let* $\mathcal{Y}_{\mathbf{a}}$ *be the set of all specializations that replace wildcards in* $\mathbf{a}$ *by any regular expression in* $\mathcal{Y}$*. Then, there are parameters* $\mathbf{u}$ *such that for each* $\mathbf{y}$ *there is a* $\mathbf{y}' \in \mathcal{Y}_{\mathbf{a}}$ *with* $f_{\mathbf{u}}(\mathbf{y}') > f_{\mathbf{u}}(\mathbf{y})$*.*

Proof Joint feature vector Ψ from Equation 6 contains two parts. The first part contains operator-nesting information over all edges in the syntax tree and the second part contains joint properties of subexpressions and the set of substrings which they generate over all nodes in the syntax tree. We construct $\mathbf{u}$ as follows: Let all weights in $\mathbf{u}$ be zero, except for the entry which weights the count of alternatives within an alternative; this entry receives any positive weight. For any string $\mathbf{a}$ that contains a wildcard symbol, by substituting the wildcard for an alternative of a wildcard and arbitrarily many other subexpressions, one can create a string $\mathbf{a}'$ that contains a wildcard within an additional alternative. Repeated application of this substitution creates arbitrarily many alternatives within alternatives and the inner product of $\mathbf{u}$ and $\Psi_{\mathbf{u}}$ can therefore become arbitrarily large. ■

Observation 1 implies that exact decoding of arbitrary decision functions $f_{\mathbf{u}}$ is not possible. However, we can follow the *under-generating* principle (Finley and Joachims, 2008) and employ a decoder that maximizes $f_{\mathbf{u}}$ over a constrained subspace that has a maximum. Observation 1 implies that the decision-function value of that maximum over the constrained space may be arbitrarily much lower than the decision-function value of some elements of the unconstrained space. But when it comes to formulating the optimization problem in Subsection 4.2.3, we will require that, for each training example, the training regular expression shall have a higher decision function value (by some margin) than the highest-scoring incorrect regular expression that is actually found by the decoder. Hence, despite Observation 1, the learning problem may produce parameters which let the constrained decoder produce the desired output.

The search space is first constrained to specializations of a maximal alignment of the input set of strings $\mathbf{x}$; see Definition 1. A maximal alignment of two strings can be determined efficiently using Hirschberg's algorithm (Hirschberg, 1975) which is an instance of dynamic programming. By contrast, finding the maximal alignment of a *set of strings* is NP-hard (Wang and Jiang, 1994); known algorithms are exponential in the number $|\mathbf{x}|$ of strings in $\mathbf{x}$. However, *progressive alignment* heuristics find an alignment of a set of strings by incrementally aligning pairs of strings. Note that the set of specializations of a maximal alignment is still generally infinitely large: each wildcard symbol can be replaced by every possible regular expression $\mathcal{Y}_{\Sigma}$. Therefore, our decoding algorithm starts by finding an approximately maximal alignment using the Hirschberg algorithm, and proceeds to construct a more constrained search space in which each wildcard symbol can be replaced only by

regular expressions over constant symbols that occur in the strings in $\mathbf{x}$ at the corresponding positions.

The definition of the constrained search space is guided by an analysis of the syntactic variants and maximum nesting depth observed in expressions written by postmasters—a detailed record can be found in the appendix; see Tables 6, 7, and 8. The space contains all specializations of the maximal alignment in which the j-th wildcard is replaced by any element from $\hat{\mathcal{Y}}_D^{M_j}$, which is constructed as follows. Firstly, $\hat{\mathcal{Y}}_D^{M_j}$ contains any subexpression that occurs within any training regular expression, and that matches the substrings of input $\mathbf{x}$ which the alignment procedure has substituted for the j-th wildcard. In addition, the alternative of all substring aligned at the j-th wildcard symbol is added. For each character-alternative expression in that set—*e.g.,* [abc]—all possible iterators and range generalizations used by postmasters are added.

Given an alignment $\mathbf{a_x} = a_0(.^*)a_1 \ldots (.^*)a_n$ of all strings in $\mathbf{x}$, the constrained search space

$$\hat{\mathcal{Y}}_{\mathbf{x},D} = \{a_0\mathbf{y}_1a_1 \ldots \mathbf{y}_na_n | \text{for all } j : \mathbf{y}_j \in \hat{\mathcal{Y}}_D^{M_j}\} \tag{7}$$

contains all specializations of $\mathbf{a_x}$ in which the j-th wildcard symbol is replaced by any element of a set $\hat{\mathcal{Y}}_D^{M_j}$, where M_j is the matching list of the j-th node in $T_{syn}^{\mathbf{a_x}}$ that is labeled with the wildcard symbol "$(.^*)$". The sets $\hat{\mathcal{Y}}_D^{M_j}$ are constructed using Algorithm 1. Each of the lines 7, 9, 10, 11, and 12 of Algorithm 1 adds at most one element to $\hat{\mathcal{Y}}_D^{M_j}$ and thus Algorithm 1 generates a finite set of possible regular expressions—hence, the search space of possible substitutions for each of the n wildcard symbols is linear in the number of subexpressions that occur in the training sample.

We now turn towards the problem of determining the highest-scoring regular expression $f_{\mathbf{w}}(\mathbf{x})$. Maximization over all regular expressions is approximated by maximization over the space defined by Equation 7:

$$\arg\max_{\mathbf{y}\in\mathcal{Y}_\Sigma} \mathbf{u}^\mathsf{T}\Psi_{\mathbf{u}}(\mathbf{x},\mathbf{y}) \approx \arg\max_{\mathbf{y}\in\hat{\mathcal{Y}}_{\mathbf{x},D}} \mathbf{u}^\mathsf{T}\Psi_{\mathbf{u}}(\mathbf{x},\mathbf{y}).$$

Due to the simple syntactic structure of the alignment and the definition of $\Psi_{\mathbf{u}}$ we can state the following theorem:

Theorem 2 *The maximization problem of finding the highest-scoring regular expression $f_{\mathbf{u}}(\mathbf{x})$ can be decomposed into independent maximization problems for each of the $\mathbf{y}_j$ that replaces the j-th wildcard in the alignment $\mathbf{a_x}$, given the alignment and the definition of $\Psi_{\mathbf{u}}$:*

$$\arg\max_{\mathbf{y}_1,\ldots,\mathbf{y}_n} f_{\mathbf{u}}(\mathbf{x}, a_0\mathbf{y}_1a_1 \ldots \mathbf{y}_na_n) = a_0\mathbf{y}_1^*a_1 \ldots \mathbf{y}_n^*a_n$$
$$\textit{with } \mathbf{y}_j^* = \arg\max_{\mathbf{y}_j\in\hat{\mathcal{Y}}_D^{M_j}} \mathbf{u}^\mathsf{T}\left(\Psi_{\mathbf{u}}(\mathbf{y}_j, M_j) + \mathbf{c}_{\mathbf{y}_j}\right).$$

Proof By its definition, $f_{\mathbf{u}}(\mathbf{x}, a_0\mathbf{y}_1a_1 \ldots \mathbf{y}_na_n) = \mathbf{u}^\mathsf{T}\Psi_{\mathbf{u}}(\mathbf{x}, a_0\mathbf{y}_1a_1 \ldots \mathbf{y}_na_n)$. Decision function Feature vector $\Psi_{\mathbf{u}}(\mathbf{x},\mathbf{y})$ decomposes linearly into a sum over the nodes and a

Algorithm 1 Constructing the decoding space

Input: Subexpressions $\mathcal{Y}_D$ and alignment $\mathbf{a_x} = a_0(.^*)a_1 \ldots (.^*)a_n$ of the strings in $\mathbf{x}$.
let $T_{syn}^{\mathbf{a_x}}$ be the syntax tree of the alignment and $v_1, \ldots, v_n$ be the nodes labeled $\Gamma_{syn}^{\mathbf{a_x}}(v_j) =$ "$(.^*)$".
for $j = 1 \ldots n$ **do**
  let $M_j = M^{\mathbf{a_x},\mathbf{x}}(v_j)$.
  Initialize $\hat{\mathcal{Y}}_D^{M_j}$ to $\{\mathbf{y} \in \mathcal{Y}_D | M_j \subseteq L(\mathbf{y})\}$
  let $x_1, \ldots, x_m$ be the elements of M_j; add $(x_1 | \ldots | x_m)$ to $\hat{\mathcal{Y}}_D^{M_j}$.
  let u be the length of the longest string and l be the length of the shortest string in M_j.
  if $[\beta \mathbf{y}_1 \ldots \mathbf{y}_k] \in \hat{\mathcal{Y}}_D^{M_j}$, where $\beta \in \Sigma^*$ and $\mathbf{y}_1 \ldots \mathbf{y}_k$ are ranges or special macros (*e.g.,* a-z, \e), then add $[\alpha \mathbf{y}_1 \ldots \mathbf{y}_k]$ to $\hat{\mathcal{Y}}_D^{M_j}$, where $\alpha \in \Sigma^*$ is the longest string that satisfies $M_j \subseteq L([\alpha \mathbf{y}_1 \ldots \mathbf{y}_k])$, if such an α exists.
  for all $[\mathbf{y}] \in \hat{\mathcal{Y}}_D^{M_j}$ **do**
    add $[\mathbf{y}]^*$ and $[\mathbf{y}]\{l, u\}$ to $\hat{\mathcal{Y}}_D^{M_j}$.
    if $l = u$, then add $[\mathbf{y}]\{l\}$ to $\hat{\mathcal{Y}}_D^{M_j}$.
    if $u \leq 1$, then add $[\mathbf{y}]?$ to $\hat{\mathcal{Y}}_D^{M_j}$.
    if $l > 0$, then add $[\mathbf{y}]^+$ to $\hat{\mathcal{Y}}_D^{M_j}$.
  end for
end for
Output $\hat{\mathcal{Y}}_D^{M_1}, \ldots, \hat{\mathcal{Y}}_D^{M_n}$.

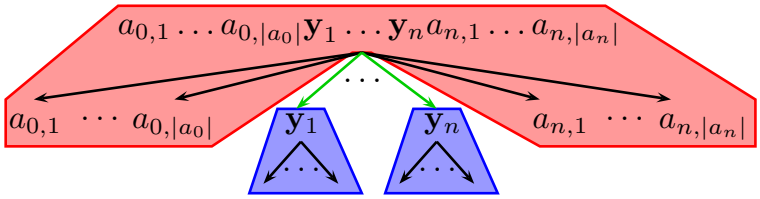


Figure 5: Structure of a syntax tree for an element of $\hat{\mathcal{Y}}_{\mathbf{x},D}$.

sum over pairs of adjacent nodes (see Equation 6). The syntax tree of an instantiation $\mathbf{y} = a_0\mathbf{y}_1a_1 \dots \mathbf{y}_na_n$ of the alignment $\mathbf{a_x}$ consists of a root node labeled as an alternating concatenation of constant strings a_j and subexpressions $\mathbf{y}_j$ (see Figure 5). This root node is connected to a layer on which constant strings $a_j = a_{j,1} \dots a_{j,|a_j|}$ and subtrees $T^{\mathbf{y}_j}_{syn}$ alternate (blue area in Figure 5). However, the terms in Equation 8 that correspond to the root node $\mathbf{y}$ and the a_j are constant for all values of the $\mathbf{y}_j$ (red area in Figure 5). Since no edges connect multiple wildcards, the feature representation of these subtrees can be decomposed into n independent summands as in Equation 9.

$$\Psi_{\mathbf{u}}(\mathbf{x}, a_0\mathbf{y}_1a_1 \dots \mathbf{y}_na_n) \tag{8}$$

$$= \begin{pmatrix} \sum_{j=1}^{n} \Lambda_{\mathbf{u}}(\mathbf{y}) \otimes \Lambda_{\mathbf{u}}(\mathbf{y}_j) + \sum_{j=0}^{n} \sum_{q=1}^{|a_j|} \Lambda_{\mathbf{u}}(\mathbf{y}) \otimes \Lambda_{\mathbf{u}}(a_{j,q}) \\ \Phi_{\mathbf{u}}(\{\mathbf{x}\}) \otimes \Lambda_{\mathbf{c}}(\mathbf{y}) + \sum_{j=0}^{n} \sum_{q=1}^{|a_j|} \Phi_{\mathbf{u}}(\{a_{j,q}\}) \otimes \Lambda_{\mathbf{u}}(a_{j,q}) \end{pmatrix}$$

$$+ \begin{pmatrix} \sum_{j=1}^{n} \sum_{(v',v'') \in E^{\mathbf{y}_j}_{syn}} \Lambda_{\mathbf{u}}(\Gamma^{\mathbf{y}_j}_{syn}(v')) \otimes \Lambda_{\mathbf{u}}(\Gamma^{\mathbf{y}_j}_{syn}(v'')) \\ \sum_{j=1}^{n} \sum_{v' \in V^{\mathbf{y}_j}_{syn}} \Phi_{\mathbf{u}}(M^{\mathbf{y}_j, M_j}(v')) \otimes \Lambda_{\mathbf{u}}(\Gamma^{\mathbf{y}_j}_{syn}(v')) \end{pmatrix}$$

$$= \begin{pmatrix} \mathbf{0} \\ \Phi_{\mathbf{u}}(\{\mathbf{x}\}) \otimes \Lambda_{\mathbf{u}}(\mathbf{y}) \end{pmatrix} + \sum_{j=0}^{n} \sum_{q=1}^{|a_i|} \begin{pmatrix} \Lambda_{\mathbf{u}}(\mathbf{y}) \otimes \Lambda_{\mathbf{u}}(a_{j,q}) \\ \Phi_{\mathbf{u}}(\{a_{j,q}\}) \otimes \Lambda_{\mathbf{u}}(a_{j,q}) \end{pmatrix}$$

$$+ \sum_{j=1}^{n} \left(\Psi_{\mathbf{u}}(\mathbf{y}_j, M_j) + \begin{pmatrix} \Lambda_{\mathbf{u}}(\mathbf{y}) \otimes \Lambda_{\mathbf{u}}(\mathbf{y}_j) \\ \mathbf{0} \end{pmatrix} \right) \tag{9}$$

Since the top-level operator of an alignment is a concatenation for any $\mathbf{y} \in \hat{\mathcal{Y}}_{\mathbf{x},D}$, we can write $\Lambda_{\mathbf{u}}(\mathbf{y})$ as a constant $\Lambda_\bullet$, defined as the output feature vector (Equation 5) of a concatenation.

Thus, the maximization over all $\mathbf{y} = a_0\mathbf{y}_1a_1 \dots \mathbf{y}_na_n$ can be decomposed into n maximization problems over

$$\mathbf{y}^*_j = \arg\max_{\mathbf{y}_j \in \hat{\mathcal{Y}}^{M_j}_D} \mathbf{u}^\mathsf{T} \left(\Psi_{\mathbf{u}}(\mathbf{y}_j, M_j) + \begin{pmatrix} \Lambda_\bullet \otimes \Lambda_{\mathbf{u}}(\mathbf{y}_j) \\ \mathbf{0} \end{pmatrix} \right)$$

which can be solved in $\mathcal{O}(n \times |\mathcal{Y}_D|)$. ■

4.2.3 Optimization Problem for Specializations of a Maximal Alignment

We will now address the process of minimizing the portion of the regularized empirical risk $\hat{R}[f_{\mathbf{w}}]$, defined in Equation 1, that depends on $\mathbf{u}$ for the ℓ_2 regularizer $\Omega_{\mathbf{c}}(\mathbf{u}) = \frac{1}{2C}||\mathbf{u}||^2$. The decision function $f_{\mathbf{w}}$ decomposes into $f_{\mathbf{u}}$ and $f_{\mathbf{v}}$; loss function $\Delta_{\mathbf{w}}$ decomposes into $\Delta_{\mathbf{u}}$ and $\Delta_{\mathbf{v}}$. While loss function $\Delta_{\mathbf{u}}$ defined in Equation 2 is not convex itself, the hinged upper bound used in Equation 1 is. Approximating a loss function by its hinged upper bound in such a way is referred to as margin-rescaling (Tsochantaridis et al., 2005). We define slack term ξ_i as this hinged loss for instance i:

$$\xi_i = \max\left\{\max_{\bar{\mathbf{y}}\neq\mathbf{y}_i}\{\mathbf{u}^\mathsf{T}(\Psi_{\mathbf{u}}(\mathbf{x}_i,\bar{\mathbf{y}}) - \Psi_{\mathbf{u}}(\mathbf{x}_i,\mathbf{y}_i)) + \Delta_{\mathbf{u}}(\mathbf{y}_i,\bar{\mathbf{y}},\mathbf{x})\}, 0\right\}. \tag{10}$$

The maximum in Equation 10 is over all $\bar{\mathbf{y}} \in \mathcal{Y}_\Sigma \setminus \{\mathbf{y}_i\}$. When the risk is rephrased as a constrained optimization problem, the maximum produces one constraint per element of $\bar{\mathbf{y}} \in \mathcal{Y}_\Sigma \setminus \{\mathbf{y}_i\}$. However, since the decoder searches only the set $\hat{\mathcal{Y}}_{\mathbf{x}_i,D}$, it is sufficient to enforce the constraints on this subset which leads to a finite search space.

When the loss is replaced by its upper bound—the slack variable ξ—and for $\Omega_{\mathbf{u}}(\mathbf{u}) = \frac{1}{2C_{\mathbf{u}}}||\mathbf{u}||^2$, the minimization of the regularized empirical risk (Equation 1) is reduced to Optimization Problem 1.

Optimization Problem 1 *Over parameters* $\mathbf{u}$*, find*

$$\mathbf{u}^* = \arg\min_{\mathbf{u},\xi} \frac{1}{2}||\mathbf{u}||^2 + \frac{C_{\mathbf{u}}}{m}\sum_{i=1}^{m}\xi_i, \text{ such that} \tag{11}$$

$$\forall i, \forall \bar{\mathbf{y}} \in \hat{\mathcal{Y}}_{\mathbf{x}_i,D}\setminus\{\mathbf{y}_i\} : \mathbf{u}^\mathsf{T}(\Psi_{\mathbf{u}}(\mathbf{x}_i,\mathbf{y}_i) - \Psi_{\mathbf{u}}(\mathbf{x}_i,\bar{\mathbf{y}})) \geq \Delta_{\mathbf{u}}(\mathbf{y}_i,\bar{\mathbf{y}},\mathbf{x}) - \xi_i, \tag{12}$$

This optimization problem is convex, since the objective (Equation 11) is convex and the constraints (Equation 12) are affine in $\mathbf{u}$. Hence, the solution is unique and can be found efficiently by cutting plane methods as Pegasos (Shalev-Shwartz et al., 2011) or SVM^{struct} (Tsochantaridis et al., 2005).

These algorithms require to identify the constraint with highest slack variable ξ_i for a given $\mathbf{x}_i$,

$$\bar{\mathbf{y}} = \arg\max_{\mathbf{y}\in\hat{\mathcal{Y}}_{\mathbf{x}_i,D}\setminus\{\mathbf{y}_i\}} \mathbf{u}^\mathsf{T}\Psi_{\mathbf{u}}(\mathbf{x}_i,\mathbf{y}) + \Delta_{\mathbf{u}}(\mathbf{y}_i,\mathbf{y},\mathbf{x}),$$

in the optimization procedure, repeatedly.

Algorithm 1 constructs the constrained search space $\hat{\mathcal{Y}}_{\mathbf{x}_i,D}$ such that $x \in L(\mathbf{y})$ for each $x \in \mathbf{x}_i$ and $\mathbf{y} \in \hat{\mathcal{Y}}_{\mathbf{x}_i,D}$. Hence, the "otherwise"-case in Equation 2 never applies within our search space. Without this case, Equations 2 and 3 decompose linearly over the nodes of the parse tree, and therefore the wildcards. Hence, $\bar{\mathbf{y}}$ can be identified by maximizing over the variables $\bar{\mathbf{y}}_j$ independently in Step 5 of Algorithm 2. Algorithm 2 finds the constraint that is violated most strongly within the constrained search space in $\mathcal{O}(n \times |\mathcal{Y}_D|)$. This ensures a polynomial execution time of the optimization algorithm. We refer to this learning procedure as *REx-SVM*.

Algorithm 2 Most strongly violated constraint

Inout: batch $\mathbf{x}$, model $f_{\mathbf{u}}$, correct output $\mathbf{y}$.
Infer alignment $\mathbf{a_x} = a_0(.^*)a_1 \dots (.^*)a_n$ for $\mathbf{x}$.
Let $T^{\mathbf{a_x}}_{syn}$ be the syntax tree of $\mathbf{a_x}$ and let $v_1, \dots, v_n$ be the nodes labeled $\Gamma^{\mathbf{a_x}}_{syn}(v_j) =$ "(.*)".
for all $j = 1 \dots n$ **do**
  Let $M_j = M^{\mathbf{a_x},\mathbf{x}}(v_j)$ and calculate the $\hat{\mathcal{Y}}_D^{M_j}$ using Algorithm 1.

$$\bar{\mathbf{y}}_j = \arg\max_{\mathbf{y}'_j \in \hat{\mathcal{Y}}_D^{M_j}} \mathbf{u}^\mathsf{T} \left(\Psi_{\mathbf{u}}(\mathbf{y}'_j, M_j) + \begin{pmatrix} \Lambda_\bullet \otimes \Lambda_{\mathbf{u}}(\mathbf{y}'_j) \\ \mathbf{0} \end{pmatrix} \right) + \Delta_{\mathbf{u}}(\mathbf{y}, a_0(.^*)a_1 \dots (.^*)a_{j-1}\mathbf{y}'_j a_j(.^*)a_{j+1} \dots (.^*)a_n, \mathbf{x})$$

end for
Let $\bar{\mathbf{y}}$ abbreviate $a_0\bar{\mathbf{y}}_1 a_1 \dots \bar{\mathbf{y}}_n a_n$
if $\bar{\mathbf{y}} = \mathbf{y}$ **then**
  Assign a value of $\bar{\mathbf{y}}'_j \in \hat{\mathcal{Y}}_D^{M_j}$ to one of the variables $\bar{\mathbf{y}}_j$ such that the smallest decrease of $f_{\mathbf{u}}(\mathbf{x}, \bar{\mathbf{y}}) + \Delta_{\text{tree}}(\mathbf{y}, \bar{\mathbf{y}})$ is obtained but the constraint $\bar{\mathbf{y}} \neq \mathbf{y}$ is enforced.
end if
Output: $\bar{\mathbf{y}}$

4.3 Learning to Extract Concise Substrings

Model $f_{\mathbf{u}}$ generates regular expressions that tend to be very specific because they are specializations of a maximal alignment of all strings in the input set $\mathbf{x}$. Human postmasters, by contrast, prefer to focus on only a characteristic part of the message for which they write a specific regular expression. In order to allow the overall model $f_{\mathbf{w}}$ to produce expressions that characterize only a part of the strings, this section focuses on a second model, $f_{\mathbf{v}}$, that selects a substring from its input string $\tilde{\mathbf{y}}$. We model $f_{\mathbf{v}}$ as a linear discriminant function with a joint feature representation $\Psi_{\mathbf{v}}$ of the input regular expression $\tilde{\mathbf{y}}$ and the output regular expression $\mathbf{y}$; decision function $f_{\mathbf{v}}$ is maximized over the set $\Pi(\tilde{\mathbf{y}})$ of all substrings of $\tilde{\mathbf{y}}$ that are themselves regular expressions:

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y} \in \Pi(\tilde{\mathbf{y}})} f_{\mathbf{v}}(\tilde{\mathbf{y}}, \mathbf{y}) = \arg\max_{\mathbf{y} \in \Pi(\tilde{\mathbf{y}})} \mathbf{v}^\mathsf{T} \Psi_{\mathbf{v}}(\tilde{\mathbf{y}}, \mathbf{y}), \tag{13}$$
$$\text{with } \Pi(\tilde{\mathbf{y}}) = \{\mathbf{y}_{\text{in}} \in \mathcal{Y}_\Sigma | \tilde{\mathbf{y}} = \mathbf{y}_{\text{pre}}\mathbf{y}_{\text{in}}\mathbf{y}_{\text{suf}} \text{ and } \mathbf{y}_{\text{pre}}, \mathbf{y}_{\text{suf}} \in \mathcal{Y}_\Sigma\}.$$

4.3.1 Joint Feature Representation for Concise Substrings

The joint feature representation $\Psi_{\mathbf{v}}(\tilde{\mathbf{y}}, \mathbf{y})$ captures structural and semantic features $\Phi_{\text{input}}(\tilde{\mathbf{y}})$ of the input regular expression $\tilde{\mathbf{y}}$, features $\Phi_{\text{output}}(\mathbf{y})$ of the output regular expression $\mathbf{y}$ and all combinations of properties of the input and output expression.

Vector $\Phi_{\text{input}}(\tilde{\mathbf{y}})$ of features of the input regular expression $\tilde{\mathbf{y}}$ includes features that indicates whether $\tilde{\mathbf{y}}$ special mail specific content like a a subject line, a "From" line, or a "Reply-To" line. A range of features test whether particular special characters are included in $\tilde{\mathbf{y}}$; other features refer to the number of subexpressions that are entailed in $\tilde{\mathbf{y}}$. The list of used features in our experiments is shown in Table 4 in the appendix.

Feature vector $\Phi_{\text{output}}(\mathbf{y})$ of the output regular expression $\mathbf{y}$ stacks up features which indicate how many subexpressions and how many words are included in the regular expression. In addition it contains features that test for special phrases that frequently occur in email batches and features that test whether words with a high spam score are included in the subject line. We identify this list of suspicious words by training a linear classifier that separates spam from non-spam emails; the list contains the 150 words which have the highest weights for the spam class. The list of features that we used in the experiments can be found in Table 5 in the appendix.

The final joint feature representation $\Psi_{\mathbf{v}}(\tilde{\mathbf{y}}, \mathbf{y})$ is defined as vector that includes the input features $\Phi_{\text{input}}(\tilde{\mathbf{y}})$, the output features $\Phi_{\text{output}}(\mathbf{y})$, and all products of an input and an output feature:

$$\Psi_{\mathbf{v}}(\tilde{\mathbf{y}}, \mathbf{y}) = \begin{pmatrix} \Phi_{\text{input}}(\tilde{\mathbf{y}}) \\ \Phi_{\text{output}}(\mathbf{y}) \\ \Phi_{\text{input}}(\tilde{\mathbf{y}}) \otimes \Phi_{\text{output}}(\mathbf{y}) \end{pmatrix}. \tag{14}$$

4.3.2 Decoding a Concise Regular Expression

At application time, the highest-scoring regular expression $\hat{\mathbf{y}}$ according to Equation 13 has to be identified. The search space $\Pi(\tilde{\mathbf{y}})$ contains all substrings of $\tilde{\mathbf{y}}$; since $\tilde{\mathbf{y}}$ is typically a very long string and calculating all features is an expensive operation, evaluating the decision function for all substrings is infeasible. Again, we follow the under-generating principle (Finley and Joachims, 2008) and constrain the search to the space $\Pi_s(\tilde{\mathbf{y}})$ contains regular expressions whose string length is at most s. Within this set, the decoder conducts an exhaustive search. One can easily observe that when the highest-scoring regular expression's string length exceeds s, then the highest-scoring regular expression of size at most s can have an arbitrarily much lower decision function value.

Observation 2 *Let* $\hat{\mathbf{y}} = \arg\max_{\mathbf{y} \in \Pi(\tilde{\mathbf{y}})} f_{\mathbf{v}}(\tilde{\mathbf{y}}, \mathbf{y})$ *and* $\hat{\mathbf{y}}_s = \arg\max_{\mathbf{y} \in \Pi(\tilde{\mathbf{y}}), |\mathbf{y}| \leq s} f_{\mathbf{v}}(\tilde{\mathbf{y}}, \mathbf{y})$. *If* $|\hat{\mathbf{y}}| > s$, *then for each number* d, *there is a parameter vector* $\mathbf{v}$ *such that* $f_{\mathbf{v}}(\tilde{\mathbf{y}}, \hat{\mathbf{y}}) > f_{\mathbf{v}}(\tilde{\mathbf{y}}, \hat{\mathbf{y}}_s) + d$.

Proof The output features of vector $\Psi_{\mathbf{v}}$ (Equation 14) include the number of constant symbols and the number of non-constant subexpressions in output expression $\mathbf{y}$. Let $\mathbf{v}$ be all zero except for these two weights which we set to $d + 1$. Then $f_{\mathbf{v}}(\tilde{\mathbf{y}}, \mathbf{y})$ is maximized by output $\hat{\mathbf{y}} = \tilde{\mathbf{y}}$. If $|\hat{\mathbf{y}}_s| < |\hat{\mathbf{y}}|$, then $\hat{\mathbf{y}}_s$ is missing at least one initial or trailing constant or non-constant symbol. By the definition of $\mathbf{v}$, decision function $f_{\mathbf{v}}(\tilde{\mathbf{y}}, \hat{\mathbf{y}}) = \mathbf{v}^{\mathsf{T}} \Psi_{\mathbf{v}}(\tilde{\mathbf{y}}, \mathbf{y}) > \mathbf{v}^{\mathsf{T}} \Psi_{\mathbf{v}}(\tilde{\mathbf{y}}, \mathbf{y}_s) + d = f_{\mathbf{v}}(\tilde{\mathbf{y}}, \hat{\mathbf{y}}_s) + d$. ■

Choosing too small a constant s can therefore lead to poor decoding results. In our experiments, we choose s to be greater than the longest regular expressions seen in the training data.

4.3.3 Optimization Problem for Concise Expressions

Training data $D = \{((\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^m$ for the overall learning problem consist of pairs of sets $\mathbf{x}_i$ of strings and corresponding regular expressions $\mathbf{y}_i$. Model $f_{\mathbf{u}}$—discussed in Section 4.2—produces intermediate expressions $\tilde{\mathbf{y}}_i$ that are specializations of a maximal alignment, before

model $f_{\mathbf{v}}(\tilde{\mathbf{y}}_i)$ gives the final predictions $\hat{\mathbf{y}}_i$. Hence, training data for model $f_{\mathbf{v}}$ naturally consists of the pairs $\{(\tilde{\mathbf{y}}_i, \mathbf{y}_i)\}_{i=1}^m$.

We will now derive an optimization problem from the portions of Equation 1 that depend on $\mathbf{v}$. Decision function $f_{\mathbf{w}}$ decomposes into $f_{\mathbf{u}} + f_{\mathbf{v}}$; loss function $\Delta_{\mathbf{w}}$ into $\Delta_{\mathbf{u}}$ and $\Delta_{\mathbf{v}}$. The regularizer decomposes, and we use the ℓ_2 regularizer for $\mathbf{v}$ as well, $\Omega_{\mathbf{s}}(\mathbf{v}) = \frac{1}{2C}||\mathbf{v}||^2$. This leads to Optimization Problem 2.

Optimization Problem 2 *Over parameters* $\mathbf{v}$*, find*

$$\begin{aligned}
&\mathbf{v}^* = \arg\min_{\mathbf{v},\xi} \frac{1}{2}||\mathbf{v}||^2 + \frac{C_{\mathbf{v}}}{m}\sum_{i=1}^m \xi_i, \text{ such that} \\
&\forall i, \forall \bar{\mathbf{y}} \in \Pi_s(\mathbf{y}_i)\backslash\{\mathbf{y}_i\} : \mathbf{v}^{\mathsf{T}}(\Psi_{\mathbf{v}}(\tilde{\mathbf{y}}_i, \mathbf{y}_i) - \Psi_{\mathbf{s}}(\tilde{\mathbf{y}}_i, \bar{\mathbf{y}})) \\
&\qquad\qquad \geq \Delta_{\mathbf{v}}(\bar{\mathbf{y}}, \mathbf{y}_i) - \xi_i, \\
&\forall i : \xi_i \geq 0.
\end{aligned}$$

Optimization Problem 2 minimizes the regularized empirical risk under the assumption that the decoder uses the restricted search space $\Pi_s(\tilde{\mathbf{y}})$ for some fixed value of the maximal string length s. We refer to the complete model

$$\begin{aligned}
\hat{\mathbf{y}} &= \arg\max_{\mathbf{y}\in\Pi(\tilde{\mathbf{y}})} \mathbf{v}^{\mathsf{T}}\Psi_{\mathbf{v}}(\tilde{\mathbf{y}}, \mathbf{y}), \\
\text{with } \tilde{\mathbf{y}} &= a_0\mathbf{y}_1^* a_1 \ldots \mathbf{y}_n^* a_n \\
\text{and } \mathbf{y}_j^* &= \arg\max_{\mathbf{y}_j\in\hat{\mathcal{Y}}_D^{M_j}} \mathbf{u}^{\mathsf{T}}\left(\Psi_{\mathbf{u}}(\mathbf{y}_j, M_j) + \left(\Lambda_\bullet \otimes \Lambda_{\mathbf{u}}(\mathbf{y}_j)\right)\right)
\end{aligned}$$

for predicting concise regular expressions as *REx-SVM*$^{\texttt{short}}$.

5. Case Study

We investigate whether postmasters accept the output of *REx-SVM* and *REx-SVM*$^{\texttt{short}}$ for blacklisting mailing campaigns during regular operations of a commercial email service. We also evaluate how accurately *REx-SVM* and *REx-SVM*$^{\texttt{short}}$ and their reference methods identify the extensions of mailing campaigns.

In order to obtain training data for the model $f_{\mathbf{u}}$ that generates a regular expression from an input batch of strings, we apply the Bayesian clustering technique of Haider and Scheffer (2009) to the stream of messages that arrive at an email service during its regular operations; the method identifies 158 mailing campaigns with a total of 12,763 messages. Postmasters of the email service write regular expressions for each batch in order to blacklist the mailing campaign; these expressions serve as labels. We will refer to this data collection as the *ESP data set*.

In order to obtain additional training data for the model $f_{\mathbf{v}}$ that selects a concise substring of a regular expression that is a specialization of the maximal alignment, we observe another 478 pairs of regular expressions with their concise subexpressions that postmasters write in order to blacklist mailing campaigns. We collected this data by using the predicted regular expression $\tilde{\mathbf{y}} = f_{\mathbf{u}}(\mathbf{x})$ for each batch of emails $\mathbf{x}$ as training observation

and the postmaster-written expression $\mathbf{y}$ as the label. We train a first-stage model $f_{\mathbf{u}}$ on the 158 labeled batches after tuning regularization parameter $C_{\mathbf{u}}$ with 10-fold cross validation. We tune the regularization parameter $C_{\mathbf{v}}$ using leave-one-out cross validation and train a global model $f_{\mathbf{v}}$ that is used in the following experiments.

5.1 Evaluation by Postmasters

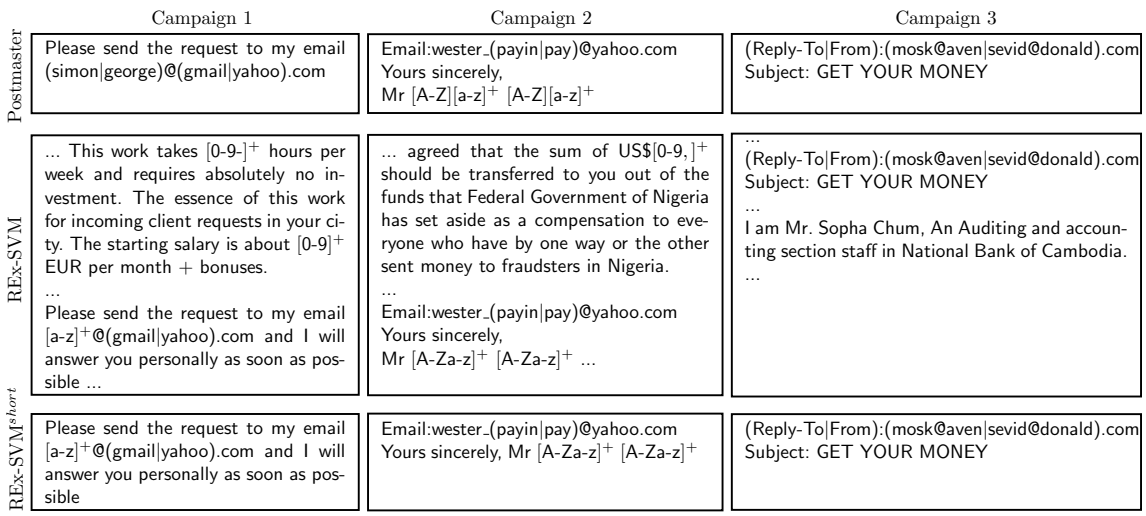


Figure 6: Regular expressions created by a postmaster and corresponding output of *REx-SVM* and *REx-SVM*$^{\texttt{short}}$.

The trained model $f_{\mathbf{u}}$ is deployed; the user interface presents newly detected batches together with the regular expressions $f_{\mathbf{u}}(\mathbf{x})$ generated by *REx-SVM* and expressions $f_{\mathbf{w}}(\mathbf{x})$ generated by *REx-SVM*$^{\texttt{short}}$ to the postmasters during regular operations of the email service. The postmasters are charged with blacklisting the campaigns with a suitable regular expression. We measure how frequently the postmasters copy the output of *REx-SVM*$^{\texttt{short}}$, copy a substring from the output of *REx-SVM*, copy but edit an output, and how frequently they choose to write an expression from scratch.

Over the course of this study, the postmasters write 153 regular expressions. They copy the exact regular expressions generated by *REx-SVM*$^{\texttt{short}}$ in 64.7% of the cases. Another 14.4% of the time, they copy a substring from the output of *REx-SVM* and use it without changes. In 7.8% of the cases, the postmasters copy and edit a substring from *REx-SVM*, and in 13.1% of the cases they write an expression from scratch. Hence, tasked with producing a regular expression that will block the mailing campaign during live operations, the postmasters prefer working with the automatically generated output to writing an expression from scratch 86.9% of the time.

To illustrate different cases, Figure 6 compares regular expressions selected by a postmaster to excerpts of regular expressions generated by *REx-SVM*, and regular expressions generated by *REx-SVM*$^{\texttt{short}}$, respectively. In the first example, *REx-SVM* over-generalizes the contact email address, and *REx-SVM*$^{\texttt{short}}$ predicts a slightly longer expression than

the postmaster prefers to select. Nevertheless, all three regular expressions characterize the extension of the mailing campaign accurately. In the second example, *REx-SVM* finds a slightly shorter but slightly more general expression for the closing signature (the term "[A-Za-z]$^+$" would allow for capital letters within the name while the term "[A-Z][a-z]$^+$" does not). The second-stage model $f_{\mathbf{v}}$ has selected the same substring that the postmaster prefers. In the third example, postmaster and *REx-SVM*^{short} agree perfectly.

The fact that postmasters are content to accept generated regular expression does not imply that they would have written the exact same rules. We now want to explore how frequently *REx-SVM*^{short} is able to produce the same regular expression that postmasters would have written. We execute leave-one-out cross validation over the regular expressions in the ESP data set. In each iteration, a new model $f_{\mathbf{u}}$ is trained on all but one regular expressions (model $f_{\mathbf{v}}$ is only trained once on different data).

We compare the output of *REx-SVM*^{short} to the held-out expression. We find that in 59.49% (94) of the cases, *REx-SVM*^{short} generates the exact regular expression written by the postmaster; in 11.39% (18) of the cases the held-out expression is a substring of the regular expression created by *REx-SVM* but distinct to the extracted expression found by *REx-SVM*^{short}. In 8.86% (14) of the cases the held-out regular expression can be obtained by modifying a substring of the string created by *REx-SVM*. In 20.25% (32) of the cases, generated and manually-written regular expression are distinct. These rates are consistent with the acceptance rates of the postmasters. When manually written and automatically generated regular expressions differ from each other, both expressions may still serve their purpose of filtering a particular batch of emails. We will explore to which extent this is the case in the next subsection.

5.2 Spam Filtering Performance

We evaluate the ability of *REx-SVM*, *REx-SVM*^{short}, and reference methods to identify the exact extension of email spam campaigns. We use the approximately maximal *alignment* of the strings determined by sequential alignment in a batch $\mathbf{x}$ as a reference method. Here, the *ReLIE* method (Li et al., 2008) serves as an additional reference. *ReLIE* takes the *alignment* as starting point of its search for a regular expression that matches the emails in the input batch and does not match any of the additional negative examples by applying a set of transformation rules. *ReLIE* receives an additional 10,000 emails that are not part of any batch as negative data, which gives it a small data advantage over *REx-SVM* and *REx-SVM*^{short}. *REx*$_{0/1}$*-SVM* is a variant of the *REx-SVM* that uses the zero-one loss instead of the loss function $\Delta_{\mathbf{u}}$ defined in Equation 2. An additional *content*-based filter employed by the provider has been trained on several million spam and non-spam emails.

Our experiments are based on two evaluation data sets. The *ESP data set* consists of the 158 batches of 12,763 emails and postmaster-written regular expressions; it is described in Section 5. In addition, we collect another 42 large spam batches with a total of 17,197 emails for which we do not have postmaster-written regular expressions. In order to be able to measure false-positive rates (the rate at which emails that are not part of a campaign are erroneously included), we use an additional 135,000 non-spam emails, also from the provider.

Additionally, we use a *public* data set that consists of 100 batches of emails extracted from the *Bruce Guenther archive*[1], containing a total of 63,512 emails. To measure false-positive rates on this public data set, we use 17,419 non-spam emails from the *Enron corpus*[2] and 76,466 non-spam emails of the *TREC corpus*[3]. The public data set is available to other researchers.

Experiments on the ESP data set are conducted as follows. We employ a constant model of $f_{\mathbf{v}}$, trained on 478 pairs of predicted expressions $\tilde{\mathbf{y}}$ and postmaster-written expressions $\mathbf{y}$. We first carry out a "leave-one-batch-out" cross-validation loop over the 158 labeled batches of the ESP data set. In each iteration, 157 batches are reserved for training of $f_{\mathbf{u}}$. On this training portion of the data, regularization parameter $C_{\mathbf{u}}$ is tuned in a nested 10-fold cross validation loop, then a model is trained on all 157 training batches. An inner loop then iterates over the size of the input batch. For each size $|\mathbf{x}|$, messages from the held-out batch are drawn into $\mathbf{x}$ at random and a regular expression $\hat{\mathbf{y}} = f_{\mathbf{w}}(\mathbf{x})$ is generated. The remaining elements of the held-out batch are used to to measure the true-positive rate of $\hat{\mathbf{y}}$, and the 135,000 non-spam emails are used to determine its false-positive rate. After that, a model is trained on all 158 labeled batches, and the evaluation iterates over the remaining 42 batches that are not labeled with a postmaster-written regular expression. For each value of $|\mathbf{x}|$, an input $\mathbf{x}$ is drawn, a prediction $\hat{\mathbf{y}}$ is generated, its true-positive rate is measured on the remaining elements of the current batch and its false-positive rate on the 135,000 non-spam messages. Standard errors are computed based on all 200 observations.

For evaluation on the public data set, parameter $C_{\mathbf{u}}$ is tuned with 10-fold cross validation and then a model is trained on all 158 labeled batches of the ESP data set. The evaluation iterates over all 100 batches of the public data set and, in an inner loop, over values of $|\mathbf{x}|$. An input set $\mathbf{x}$ is drawn at random from the current batch, the true-positive rate of $\hat{\mathbf{y}} = f_{\mathbf{w}}(\mathbf{x})$ is measured on the remaining elements of the current batch and the false-positive rate of $\hat{\mathbf{y}}$ is measured on the Enron and TREC emails.

Figure 7 shows the true- and false-positive rates for all methods on both data sets. The horizontal axis displays the number of emails in the input batch $\mathbf{x}$. Error bars indicate the standard error. The true-positive rate measures the proportion of a batch that is recognized while the false-positive rate counts emails that match a regular expression although they are not an element of the corresponding campaign. The *alignment* has the highest true-positive rate and a high false-positive rate because it is the most general bound of the decoder's search space. *ReLIE* only has to carry out very few transformation steps until no negative examples are covered—in some cases none at all. Consequently, it has similarly high true- and false-positive rates. *REx-SVM* and *REx-SVM*^{short} attain a slightly lower true-positive rate, and a substantially lower false-positive rate. The false-positive rates of *REx-SVM*, *REx*$_{0/1}$*-SVM*, and *REx-SVM*^{short} lie more than an order of magnitude below the rate of the commercial *content*-based spam filter employed by the email service provider. The zero-one loss leads to comparable false-positive but lower true-positive rates, rendering the loss function $\Delta_{\mathbf{u}}$ preferable to the zero-one loss. The true-positive rate of *REx-SVM*^{short} is significantly higher than the true-positive rate of *REx-SVM* for small sizes of the input batch; it requires only very few input strings in order to generate regular expressions which

1. `http://untroubled.org/spam/`
2. `http://www.cs.cmu.edu/~enron/`
3. `http://trec.nist.gov/data/spam.html`

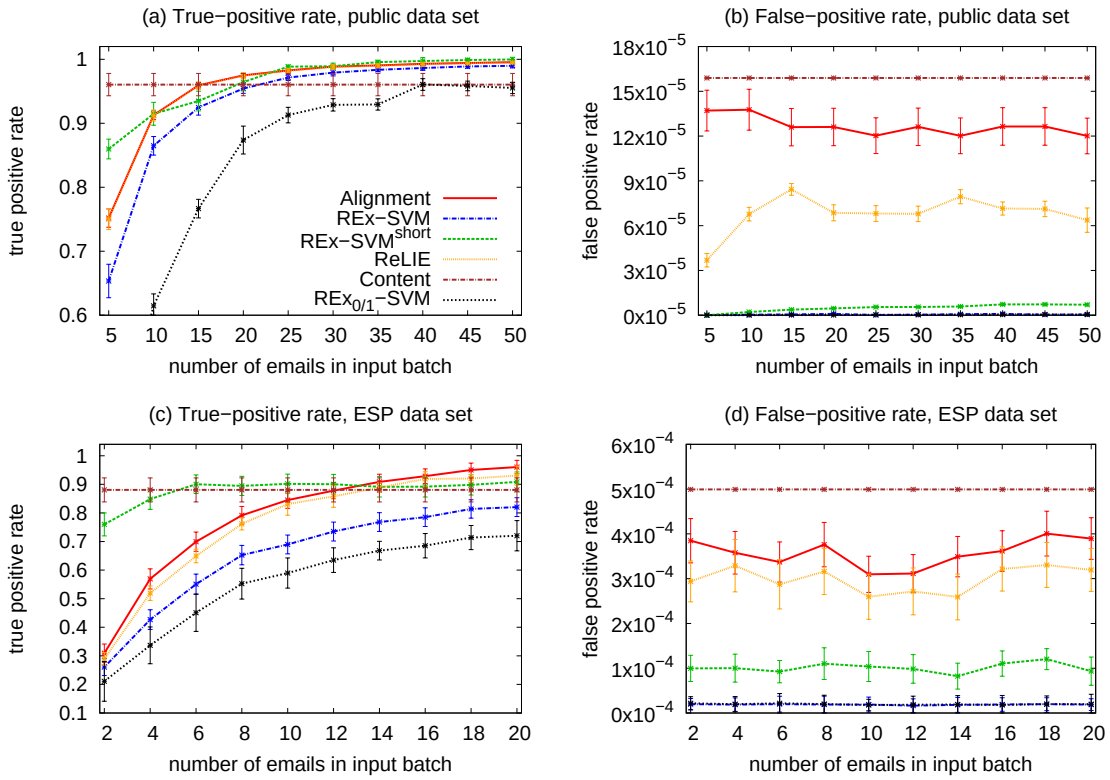


Figure 7: True-positive and false-positive rates over the number of used emails in the input batch $\mathbf{x}$ for the public and ESP data sets.

can be used to describe nearly the entire extension of a batch at a very low false-positive rate.

Finally, we determine the risk of the studied methods producing a regular expression that causes at least one false-positive match of an email which does not belong to the batch. *REx-SVM*'s risk of producing a regular expression that incurs at least one false-positive match is 2.5%; for *REx-SVM*^{short}, this risk is 3.7%; for *alignment*, the risk is 6.3%, and for *ReLIE*, it is 5.1%.

5.3 Learning Curves, Execution Time

We study learning curves of the loss functions of *REx-SVM* and *REx-SVM*^{short}. Figure 8 (a) shows the average loss $\Delta_{\mathbf{u}}$ based on cross validation with one batch held out, as a function of the number of training batches. The "minimum loss" baseline shows the smallest possible loss within the constrained search space; it visualizes how much constraining the search space contributes to the overall loss. This value is obtained by an altered search procedure that minimizes the loss function between prediction and the postmaster-written regular expression instead of the decision function. This loss-minimizing expression has a

lower decision function value than the predicted regular expression; the difference between minimum loss and the loss of *REx-SVM* and $REx_{0/1}$-*SVM*, respectively, can be attributed to imperfections of the model. Figure 8 (a) also shows the loss of the *alignment*. This loss serves as an upper bound and visualizes how much the parameterized models contribute to minimizing the error. For completeness, Figure 10 in the appendix shows the learning curves on the training data.

Figure 8 (b) shows the average loss $\Delta_{\mathbf{v}}$ based on 10 fold cross validation and the average loss on the training data. The impact of the regularization parameters $C_{\mathbf{u}}$ and $C_{\mathbf{v}}$ is shown in Figure 11 in the appendix.

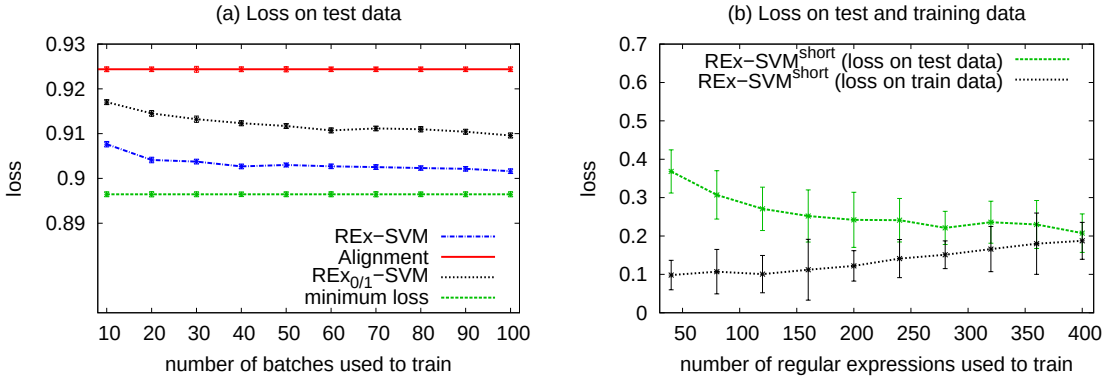


Figure 8: (a) Loss $\Delta_{\mathbf{u}}$ of model $f_{\mathbf{u}}$ on the test data (left Figure). (b) Loss $\Delta_{\mathbf{v}}$ of model $f_{\mathbf{v}}$ on the training and test data. Error bars indicate standard errors.

Table 5.3 measures how much *REx-SVM*^{short} reduces the length of the expressions produced by *REx-SVM*. We can conclude that *REx-SVM*^{short} reduces the length of the output of *REx-SVM* by an average of 92%.

Method	mean	standard error
REx-SVM	2141	2063
REx-SVM^{short}	95	92

Table 1: Number of characters in automatically-generated regular expressions.

The execution time for learning is consistent with prior findings of between linear and quadratic for the SVM optimization process—see Figure 9(a). Figure 9 (b) shows the execution time of the decoder that generates a regular expression for input batch $\mathbf{x}$ at application time. *ReLIE* does not require training.

In order to use regular expressions to blacklist email spam, the email service provider's infrastructure has to continuously match all active regular expressions against the stream of incoming emails. This acceptor is implemented as a deterministic finite-state automaton.

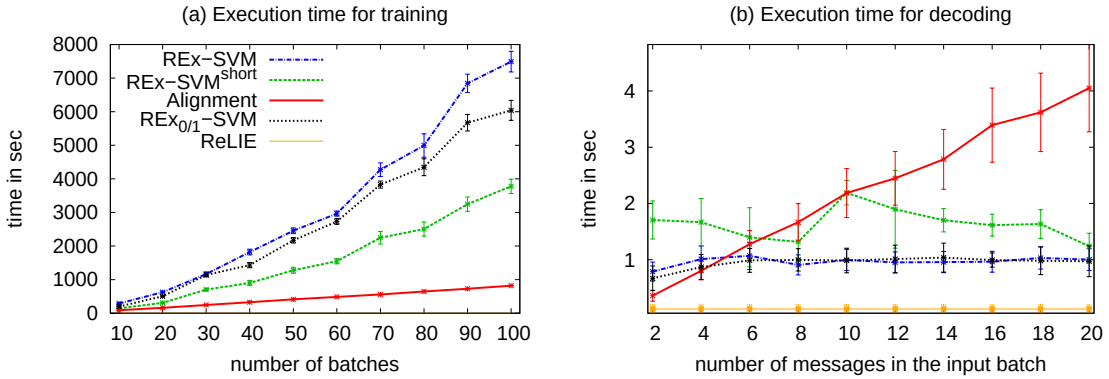


Figure 9: Execution time for training a model (a) and decoding a regular expression at application time (b).

The automaton has to be kept in main memory, and therefore the number of states determines the number of regular expressions that can be searched for in parallel. Table 2 shows the average number of states of an acceptor, generated by the method of Dubé and Feeley (2000) from the regular expressions of *REx-SVM* and *REx-SVM*^{short}. The average number of states of regular expressions by *REx-SVM*^{short} is close to the average number of states of expressions written by a human postmaster, while *alignment*, *ReLIE*, and *REx-SVM* require impractically large accepting automata.

Method	mean	median	standard error
alignment	5709	4059	389.1
REx-SVM	5473	2995	520.8
ReLIE	5632	3587	465.9
REx-SVM^{short}	72	69	1.8
postmaster	68	48	4.6

Table 2: Number of states of an accepting finite-state automaton.

6. Related Work

Gold (1967) shows that it is impossible to exactly identify any regular language from finitely many positive examples. In his framework, a learner makes a conjecture after each new positive example; only finitely many initial conjectures may be incorrect. Our notion of minimizing an expected difference between conjecture and target language over a distribution of input strings reflects a more statistically-inspired notion of learning. Also, in our problem setting the learner has access to pairs of sets of strings and corresponding regular expressions.

Most work of identification of regular languages focuses on learning automata (Denis, 2001; Parekh and Honavar, 2001; Clark and Thollard, 2004). Since regular languages are

accepted by finite automata, the problems of learning regular languages and learning finite automata are tightly coupled. However, a compact regular language may have an accepting automaton with a large number of states and, analogously, transforming compact automata into regular expressions can lead to lengthy terms that do not lend themselves to human comprehension (Fernau, 2009).

Positive learnability results can be obtained for restricted classes of deterministic finite automata with positive examples (Angluin, 1978; Abe and Warmuth, 1990); for instance expressions in which each symbol occurs at most k times (Bex et al., 2008), disjunction-free expressions (Brāzma, 1993), and disjunctions of left-aligned disjunction-free expressions (Fernau, 2009) have been studied. These approaches aim only at the identification of a target language. By contrast, here the structural resemblance of the conjecture to a target regular expression is integral part of the problem setting. This also makes it necessary to account for the broader syntactic spectrum of regular expressions.

Xie et al. (2008) use regular expressions to detect URLs in spam batches and develop a spam filter with low false-positive rate. The *ReLIE*-algorithm (Li et al., 2008) (used as a reference method in our experiments) learns regular expressions from positive and negative examples given an initial expression by applying a set of transformation rules as long as this improves the separation of positive and negative examples. Brauer et al. (2011) develop an algorithm that builds a data structure of commonalities of several aligned strings and transforms these strings into a specific regular expression. Because of a high data overhead, their algorithm works best for short strings, such as telephone numbers and names of software products.

Structured output spaces are a flexible tool for a wide array of problem settings, including sequence labeling, sequence alignment, and natural language parsing (Tsochantaridis et al., 2005). In our problem setting we are interested in predicting a structured object, i.e. a regular expression. To solve problems with structured output spaces an extension of the support vector machines (SVMs, Vapnik, 1998) can be used. Such structural SVMs were used to solve a several number of prediction tasks ranging from classification with taxonomies, label sequence learning, sequence alignment to natural language parsing (Tsochantaridis et al., 2005). The problem of detecting message campaigns in the stream of emails has been addressed with structured output spaces based on manually grouped training messages (Haider et al., 2007), and with graphical models without the need for labeled training data (Haider and Scheffer, 2009).

Our problem setting and method differ from all prior work on learning regular expressions in their objective criterion and training data. Unlike in prior work, the learner in our setting has access to additional labeled data in the form of pairs of a set of strings and a corresponding regular expressions. At the same time, the learner's goal is not just to find an expression that identifies an extension of strings, but to find *the* expression which the process that has labeled the training data would most likely generate. This implies that the learner has to model the labeler's preference of using specific syntactic constructs in a specific syntactic context and for specific matching substrings.

7. Conclusions

Complementing the language-identification paradigm, we address the problem of learning to map a set of strings to a concise regular expression that resembles an expression which a human would have written. Training data consists of batches of strings and corresponding regular expressions. We phrase this problem as a two-staged learning problem with structured output spaces and engineer appropriate loss functions. We devise a first-stage decoder that searches a space of specializations of a maximal alignment of the input strings. We devise a second-stage decoder that searches for a substring of the first-stage result. We derive optimization problems for both stages.

From our case study, we conclude that *REx-SVM*$^{\texttt{short}}$ frequently predicts the exact regular expression that a postmaster would have written. In other cases, it generates an expression that postmasters accept without or with small modifications. Regarding their accuracy for the problem of filtering email spam, we conclude that *REx-SVM* and *REx-SVM*$^{\texttt{short}}$ give a high true-positive rate at a false-positive rate that is an order of magnitude lower than that of a commercial content-based filter. *REx-SVM*$^{\texttt{short}}$ attains a higher true-positive rate, in particular for small input batches. *REx-SVM*$^{\texttt{short}}$ generates regular expressions that can be accepted by a finite-state automaton that has just slightly more states than an accepting automaton for regular expressions written by a human postmaster. *REx-SVM* and all reference methods, by contrast, can only be accepted by impractically large finite-state automata. *REx-SVM*$^{\texttt{short}}$ is being used by a commercial email service provider and complements content-based and IP-address based filtering.

Acknowledgments

This work was funded by a grant from STRATO AG. We would like to thank the anonymous reviewers for their helpful comments.

References

N. Abe and M. K. Warmuth. On the computational complexity of approximating distributions by probabilistic automata. In *Proceedings of the Conference on Learning Theory*, pages 52–66, 1990.

D. Angluin. On the complexity of minimum inference of regular sets. *Information and Control*, 39(3):337–350, 1978.

G. Bex, W. Gelade, F. Neven, and S. Vansummeren. Learning deterministic regular expressions for the inference of schemas from XML data. In *Proceeding of the International World Wide Web Conference*, pages 825–834, 2008.

F. Brauer, R. Rieger, A. Mocan, and W.M. Barczynski. Enabling information extraction by inference of regular expressions from sample entities. In *Proceedings of the Conference on Information and Knowledge Management*, pages 1285–1294. ACM, 2011. ISBN 978-1-4503-0717-8.

A. Brāzma. Efficient identification of regular expressions from representative examples. In *Proceedings of the Annual Conference on Computational Learning Theory*, pages 236–242, 1993.

N. Cesa-Bianchi, C. Gentile, and L. Zaniboni. Incremental algorithms for hierarchical classification. *Machine Learning*, 7:31–54, 2006.

A. Clark and F. Thollard. PAC-learnability of probabilistic deterministic finite state automata. *Journal of Machine Learning Research*, 5:473–497, 2004.

F. Denis. Learning regular languages from simple positive examples. *Machine Learning*, 44:27–66, 2001.

D. Dubé and M. Feeley. Efficiently building a parse tree from a regular expression. *Acta Informatica*, 37(2):121–144, 2000.

H. Fernau. Algorithms for learning regular expressions from positive data. *Information and Computation*, 207(4):521–541, 2009.

T. Finley and T. Joachims. Training structural SVMs when exact inference is intractable. In *Proceedings of the International Conference on Machine Learning*, 2008.

E. M. Gold. Language identification in the limit. *Information and Control*, 10:447–474, 1967.

P. Haider and T. Scheffer. Bayesian clustering for email campaign detection. In *Proceedings of the International Conference on Machine Learning*, 2009.

P. Haider, U. Brefeld, and T. Scheffer. Supervised clustering of streaming data for email batch detection. In *Proceedings of the International Conference on Machine Learning*, 2007.

D. Hirschberg. A linear space algorithm for computing maximal common subsequences. *Communications of the ACM*, 18(6):341–343, 1975.

Y. Li, R. Krishnamurthy, S. Raghavan, S. Vaithyanathan, and H. V. Jagadish. Regular expression learning for information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 21–30, 2008.

R. Parekh and V. Honavar. Learning DFA from simple examples. *Machine Learning*, 44: 9–35, 2001.

P. Prasse, C. Sawade, N. Landwehr, and T. Scheffer. Learning to identify regular expressions that describe email campaigns. In *Proceedings of the International Conference on Machine Learning*, 2012.

S. Shalev-Shwartz, Y. Singer, N. Srebro, and A. Cotter. Pegasos: primal estimated subgradient solver for SVM. *Mathematical Programming*, 127(1):1–28, 2011.

I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6:1453–1484, 2005.

V. Vapnik. *Statistical Learning Theory*. Wiley, 1998.

L. Wang and T. Jiang. On the complexity of multiple sequence alignment. *Journal of Computational Biology*, 1(4):337–348, 1994.

Y. Xie, F. Yu, K. Achan, R. Panigrahy, G. Hulten, and I. Osipkov. Spamming botnets: signatures and characteristics. In *Proceedings of the ACM SIGCOMM Conference*, pages 171–182, 2008.

Appendix A

A.1 Syntax and Semantics of Regular Expressions

Definition 3 (Regular Expressions) The set $\mathcal{Y}_\Sigma$ of regular expressions over an ordered alphabet Σ is recursively defined as follows.

- Every $\mathbf{y}_j \in \Sigma \cup \{\epsilon, ., \backslash\mathsf{S}, \backslash\mathsf{e}, \backslash\mathsf{w}, \backslash\mathsf{d}\}$, every range $\mathbf{y}_j = l_{min}\text{–}l_{max}$, where $l_{min}, l_{max} \in \Sigma$ and $l_{min} < l_{max}$, and their disjunction $[\mathbf{y}_1 \dots \mathbf{y}_k]$ are regular expressions.
- If $\mathbf{y}_1, \dots, \mathbf{y}_k \in \mathcal{Y}_\Sigma$ are regular expressions, so are the concatenation $\mathbf{y} = \mathbf{y}_1 \dots \mathbf{y}_k$, the disjunction $\mathbf{y} = \mathbf{y}_1 | \dots | \mathbf{y}_k$, $\mathbf{y} = \mathbf{y}_1?$, $\mathbf{y} = (\mathbf{y}_1)$, and the repetitions $\mathbf{y} = \mathbf{y}_1^*$, $\mathbf{y} = \mathbf{y}_1^+$, $\mathbf{y} = \mathbf{y}_1\{l\}$, and $\mathbf{y} = \mathbf{y}_1\{l, u\}$, where $l, u \in \mathbb{N}$ and $l \leq u$.

We now define the syntax tree, the parse tree, and the matching lists for a regular expression $\mathbf{y}$ and a string $x \in \Sigma^*$. The shorthand $(\mathbf{y} \to T_1, \dots, T_k)$ denotes the tree $T = (V, E, \Gamma, \leq)$ with root node $v_0 \in V$ labeled with $\Gamma(v_0) = \mathbf{y}$ and subtrees $T_1, \dots, T_k$. The order $\leq$ maintains the subtree orderings $\leq_i$ and defines the root node as the minimum over the set V and $v' \leq v''$ for all $v' \in V_i$ and $v'' \in V_j$, where $i < j$.

Definition 4 (Syntax Tree) The abstract syntax tree $T_{syn}^{\mathbf{y}}$ for a regular expression $\mathbf{y}$ is recursively defined as follows. Let $T_{syn}^{\mathbf{y}_j} = (V_{syn}^{\mathbf{y}_j}, E_{syn}^{\mathbf{y}_j}, \Gamma_{syn}^{\mathbf{y}_j}, \leq_{syn}^{\mathbf{y}_j})$ be the syntax tree of the subexpression $\mathbf{y}_j$.

- If $\mathbf{y} \in \Sigma \cup \{\epsilon, ., \backslash\mathsf{S}, \backslash\mathsf{e}, \backslash\mathsf{w}, \backslash\mathsf{d}\}$, or if $\mathbf{y} = l_{min}\text{–}l_{max}$, where $l_{min}, l_{max} \in \Sigma$, we define $T_{syn}^{\mathbf{y}} = (\mathbf{y} \to \emptyset)$.
- If $\mathbf{y} = (\mathbf{y}_1)$, where $\mathbf{y}_1 \in \mathcal{Y}_\Sigma$, we define $T_{syn}^{\mathbf{y}} = T_{syn}^{\mathbf{y}_1}$.
- If $\mathbf{y} = \mathbf{y}_1^*$, $\mathbf{y} = \mathbf{y}_1^+$, $\mathbf{y} = \mathbf{y}_1\{l, u\}$, or if $\mathbf{y} = \mathbf{y}_1\{l\}$, where $\mathbf{y}_1 \in \mathcal{Y}_\Sigma$, $l, u \in \mathbb{N}$, and there exist no $\mathbf{y}', \mathbf{y}'' \in \mathcal{Y}_\Sigma$ such that $\mathbf{y}_1 = \mathbf{y}'|\mathbf{y}''$ or $\mathbf{y}_1 = \mathbf{y}'\mathbf{y}''$, we define $T_{syn}^{\mathbf{y}} = (\mathbf{y} \to T_{syn}^{\mathbf{y}_1})$.
- If $\mathbf{y} = \mathbf{y}_1 \dots \mathbf{y}_k$, where $\mathbf{y}_j \in \mathcal{Y}_\Sigma$, and there exist no $\mathbf{y}', \mathbf{y}'' \in \mathcal{Y}_\Sigma$ such that $\mathbf{y}_j = \mathbf{y}'|\mathbf{y}''$ or $\mathbf{y}_j = \mathbf{y}'\mathbf{y}''$, we define $T_{syn}^{\mathbf{y}} = (\mathbf{y} \to T_{syn}^{\mathbf{y}_1}, \dots, T_{syn}^{\mathbf{y}_k})$.
- If $\mathbf{y} = \mathbf{y}_1 | \dots | \mathbf{y}_k$, where $\mathbf{y}_j \in \mathcal{Y}_\Sigma$, and there exist no $\mathbf{y}', \mathbf{y}'' \in \mathcal{Y}_\Sigma$ such that $\mathbf{y}_j = \mathbf{y}'|\mathbf{y}''$, or if $\mathbf{y} = [\mathbf{y}_1 \dots \mathbf{y}_k]$ and there exist no $\mathbf{y}', \mathbf{y}'' \in \mathcal{Y}_\Sigma$ such that $\mathbf{y}_j = \mathbf{y}'\mathbf{y}''$, we define $T_{syn}^{\mathbf{y}} = (\mathbf{y} \to T_{syn}^{\mathbf{y}_1}, \dots, T_{syn}^{\mathbf{y}_k})$.

Definition 5 (Parse Tree and Matching List) Given a syntax tree $T^{\mathbf{y}}_{syn} = (V^{\mathbf{y}}_{syn}, E^{\mathbf{y}}_{syn}, \Gamma^{\mathbf{y}}_{syn}, \leq^{\mathbf{y}}_{syn})$ of a regular expression $\mathbf{y}$ with nodes $v \in V^{\mathbf{y}}_{syn}$ and a string $x \in L(\mathbf{y})$, a parse tree $T^{\mathbf{y},x}_{par}$ and the matching lists $M^{\mathbf{y},x}(v)$ for each $v \in V^{\mathbf{y}}_{syn}$ are recursively defined as follows. Let $T^{\mathbf{y}_j,x}_{par} = (V^{\mathbf{y}_j,x}_{par}, E^{\mathbf{y}_j,x}_{par}, \Gamma^{\mathbf{y}_j,x}_{par}, \leq^{\mathbf{y}_j,x}_{par})$ be the parse tree and $T^{\mathbf{y}_j}_{syn} = (V^{\mathbf{y}_j}_{syn}, E^{\mathbf{y}_j}_{syn}, \Gamma^{\mathbf{y}_j}_{syn}, \leq^{\mathbf{y}_j}_{syn})$ the syntax tree of the subexpression $\mathbf{y}_j$.

- If $\mathbf{y} = x$ and $x \in \Sigma \cup \{\epsilon\}$, we define
  $M^{\mathbf{y},x}(v_0) = \{x\}$ and
  $T^{\mathbf{y},x}_{par} = (\mathbf{y} \to \emptyset)$.

- If $\mathbf{y} = .$ and $x \in \Sigma$,
  $\mathbf{y} = l_{min}\text{–}l_{max}$ and $l_{min} \leq x \leq l_{max}$, or if
  $\mathbf{y} \in \{\backslash\mathsf{S}, \backslash\mathsf{w}, \backslash\mathsf{e}, \backslash\mathsf{d}\}$ and x is either a non-whitespace character (everything but spaces, tabs, and line breaks), a word character (letters, digits, and underscores), a character in $\{., -, \#, +\}$ or a word character, or a digit, respectively, we define
  $M^{\mathbf{y},x}(v) = \{x\}$ for all $v \in V^{\mathbf{y}}_{syn}$ and
  $T^{\mathbf{y},x}_{par} = (\mathbf{y} \to T^{x,x}_{par})$.

- If $\mathbf{y} = (\mathbf{y}_1)$ and $x \in \Sigma^*$, we define
  $M^{\mathbf{y},x}(v) = M^{\mathbf{y}_1,x}(v)$ for all $v \in V^{\mathbf{y}}_{syn}$ and
  $T^{\mathbf{y},x}_{par} = T^{\mathbf{y}_1,x}_{par}$

- If $\mathbf{y} = \mathbf{y}_1^*$, $x = x_1 \dots x_k$, and $k \geq 0$, or if
  $\mathbf{y} = \mathbf{y}_1^+$, and $k > 0$, or if
  $\mathbf{y} = \mathbf{y}_1\{l, u\}$, and $l \leq k \leq u$, or if
  $\mathbf{y} = \mathbf{y}_1\{l\}$, and $k = l$,
  where $x_i \in \Sigma^+$, and there exist no $\mathbf{y}', \mathbf{y}'' \in \mathcal{Y}_\Sigma$ such that $\mathbf{y}_1 = \mathbf{y}'|\mathbf{y}''$ or $\mathbf{y}_1 = \mathbf{y}'\mathbf{y}''$, we define
  $M^{\mathbf{y},x}(v) = \begin{cases} \{x\} & \text{, if } v = v_0 \\ \bigcup_{i=1}^{k} M^{\mathbf{y}_1,x_i}(v) & \text{, if } v \in V^{\mathbf{y}_1}_{syn} \end{cases}$, and
  $T^{\mathbf{y},x}_{par} = (\mathbf{y} \to T^{\mathbf{y}_1,x_1}_{par}, \dots, T^{\mathbf{y}_1,x_k}_{par})$.

- If $\mathbf{y} = \mathbf{y}_1 \dots \mathbf{y}_k$, $x = x_1 \dots x_k$,
  where $x_i \in \Sigma^*$, and there exist no $\mathbf{y}', \mathbf{y}'' \in \mathcal{Y}_\Sigma$ such that $\mathbf{y}_j = \mathbf{y}'|\mathbf{y}''$ or $\mathbf{y}_j = \mathbf{y}'\mathbf{y}''$, we define
  $M^{\mathbf{y},x}(v) = \begin{cases} \{x\} & \text{, if } v = v_0 \\ M^{\mathbf{y}_j,x_i}(v) & \text{, if } v \in V^{\mathbf{y}_j}_{syn} \end{cases}$, and
  $T^{\mathbf{y},x}_{par} = (\mathbf{y} \to T^{\mathbf{y}_1,x_1}_{par}, \dots, T^{\mathbf{y}_k,x_k}_{par})$.

- If $\mathbf{y} = \mathbf{y}_1| \dots |\mathbf{y}_k$, $x \in \Sigma^*$
  and there exist no $\mathbf{y}', \mathbf{y}'' \in \mathcal{Y}_\Sigma$ such that $\mathbf{y}_j = \mathbf{y}' \mid \mathbf{y}''$, or if
  $\mathbf{y} = [\mathbf{y}_1 \dots \mathbf{y}_k]$, $x \in \Sigma^+$
  and there exist no $\mathbf{y}', \mathbf{y}'' \in \mathcal{Y}_\Sigma$ such that $\mathbf{y}_j = \mathbf{y}'\ \mathbf{y}''$, we define
  $M^{\mathbf{y},x}(v) = \begin{cases} \{x\} & \text{, if } v = v_0 \\ M^{\mathbf{y}_j,x}(v) & \text{, if } v \in V^{\mathbf{y}_j}_{syn} \text{, and} \\ \emptyset & \text{, otherwise} \end{cases}$
  $T^{\mathbf{y},x}_{par} = (\mathbf{y} \to T^{\mathbf{y}_j,x}_{par})$.

If $x \notin L(\mathbf{y})$, that is, no parse tree can be derived by the specification above, the empty sets $M^{\mathbf{y},x}(v) = \emptyset$ for all $v \in V_{syn}^{\mathbf{y}}$ and $T_{par}^{\mathbf{y},x} = \emptyset$ are returned. Otherwise, we denote the set of all parse trees and the unions of all matching lists for each $v \in V_{syn}^{\mathbf{y}}$ satisfying Definition 5 by $\mathcal{T}_{par}^{\mathbf{y},x}$ and $\mathcal{M}^{\mathbf{y},x}(v)$, respectively. Finally, the matching list $M^{\mathbf{y},\mathbf{x}}(v)$ for a set of strings $\mathbf{x}$ for node $v \in V_{syn}^{\mathbf{y}}$ is defined as $M^{\mathbf{y},\mathbf{x}}(v) = \bigcup_{x \in \mathbf{x}} \mathcal{M}^{\mathbf{y},x}(v)$.

A.2 Joint Feature Representations

The list of binary and continuous features $\Psi_{\mathbf{u}}$ used to train model $f_{\mathbf{u}}$ is shown in Table 3. The input and output features $\Psi_{\mathbf{v}}$ for model $f_{\mathbf{v}}$ are shown in Table 4 and 5, respectively. The set S_{spam} is defined as follows: We train a linear classifier that separates spam emails from non-spam emails on the ESP data set, using a bag of words representation. We construct the set S_{spam} as the 150 words that have the highest weights for the class spam.

Feature	Description
$[\![\varepsilon \in M]\!]$	Matching list contains the empty string?
$[\![\forall \mathbf{x} \in M : \lvert\mathbf{x}\rvert = 1]\!]$	All elements of the matching list have the length one?
$[\![\exists i \in \mathbb{N}, \forall \mathbf{x} \in M : \lvert\mathbf{x}\rvert = i]\!]$	All elements of the matching list have the same length?
$\frac{\lvert\Sigma_M \cap \{A,\ldots,Z\}\rvert}{26}$	Portion of characters A–Z in the matching list
$\frac{\lvert\Sigma_M \cap \{a,\ldots,z\}\rvert}{26}$	Portion of characters a–z in the matching list
$\frac{\lvert\Sigma_M \cap \{0,\ldots,9\}\rvert}{10}$	Portion of characters 0–9 in the matching list
$\frac{\lvert\Sigma_M \cap \{A,\ldots,F\}\rvert}{6}$	Portion of characters A–F in the matching list
$\frac{\lvert\Sigma_M \cap \{a,\ldots,f\}\rvert}{6}$	Portion of characters a–f in the matching list
$\frac{\lvert\Sigma_M \cap \{G,\ldots,Z\}\rvert}{20}$	Portion of characters G–Z in the matching list
$\frac{\lvert\Sigma_M \cap \{g,\ldots,z\}\rvert}{20}$	Portion of characters g–z in the matching list
$[\![\forall x \in \Sigma_M : x \notin \{A,\ldots,Z\}\ ]\!]$	No characters of A–Z in the matching list?
$[\![\forall x \in \Sigma_M : x \notin \{a,\ldots,z\}\ ]\!]$	No characters of a–z in the matching list?
$[\![\forall x \in \Sigma_M : x \notin \{0,\ldots,9\}\ ]\!]$	No characters of 0–9 in the matching list?
$[\![\forall x \in \Sigma_M : x \notin \{a,\ldots,f\}\ ]\!]$	No characters of a–f in the matching list?
$[\![\forall x \in \Sigma_M : x \notin \{A,\ldots,F\}\ ]\!]$	No characters of A–F in the matching list?
$[\![\lvert\Sigma_M \cap \{-,/,?,=,.,@,:\}\rvert > 0]\!]$	Matching list contains URL/Email characters?
$[\![\forall \mathbf{x} \in M : \lvert\mathbf{x}\rvert \geq 1 \wedge \lvert\mathbf{x}\rvert \leq 5]\!]$	Length of strings in the matching list is between 1 and 5?
$[\![\forall \mathbf{x} \in M : \lvert\mathbf{x}\rvert \geq 6 \wedge \lvert\mathbf{x}\rvert \leq 10]\!]$	Length of strings in the matching list is between 5 and 10?
$[\![\forall \mathbf{x} \in M : \lvert\mathbf{x}\rvert \geq 11 \wedge \lvert\mathbf{x}\rvert \leq 20]\!]$	Length of strings in the matching list is between 10 and 20?
$[\![\forall \mathbf{x} \in M : \lvert\mathbf{x}\rvert > 20]\!]$	Length of strings in the matching list is higher than 20?
$[\![\lvert M\rvert = 0]\!]$	Matching list is empty?

Table 3: Features for model $f_{\mathbf{u}}$.

A.3 Additional Experimental Results

Figure 10 shows the average loss $\Delta_{\mathbf{u}}$ on the training data as a function of the sample size. The corresponding loss on the test data can be seen in Figure 8 (a).

Feature	Description
$[\![0 \leq$ constant symbols in $\tilde{\mathbf{y}} < 568]\!]$	Number of constant symbols that are arguments of the top-most concatenation is less than 568
$[\![568 \leq$ constant symbols in $\tilde{\mathbf{y}} < 1032]\!]$	... between 568 and 1031
$[\![1032 \leq$ constant symbols in $\tilde{\mathbf{y}} < 1724]\!]$	... between 1032 and 1723
$[\![1724 \leq$ constant symbols in $\tilde{\mathbf{y}} < 2748]\!]$	... between 1724 and 2747
$[\![2748 \leq$ constant symbols in $\tilde{\mathbf{y}}]\!]$	... 2748 or higher
$[\![0 \leq$ non-constant arguments in $\tilde{\mathbf{y}} < 48]\!]$	Number of non-constant arguments of the top-level concatenation is less than 48
$[\![48 \leq$ non-constant arguments in $\tilde{\mathbf{y}} < 77]\!]$	... between 48 and 76
$[\![77 \leq$ non-constant arguments in $\tilde{\mathbf{y}} < 133]\!]$	... between 77 and 132
$[\![133 \leq$ non-constant arguments in $\tilde{\mathbf{y}} < 246]\!]$	... between 133 and 245
$[\![246 \leq$ non-constant arguments in $\tilde{\mathbf{y}}]\!]$	... 246 or higher
$[\![\tilde{\mathbf{y}}$ contains Latin characters$]\!]$	
$[\![\tilde{\mathbf{y}}$ contains Greek characters$]\!]$	
$[\![\tilde{\mathbf{y}}$ contains Russian characters$]\!]$	
$[\![\tilde{\mathbf{y}}$ contains Asian characters$]\!]$	
$[\![\tilde{\mathbf{y}}$ contains "subject:"$]\!]$	Expression refers to a subject line
$[\![\tilde{\mathbf{y}}$ contains "from:"$]\!]$	Refers to a sender address
$[\![\tilde{\mathbf{y}}$ contains "to:"$]\!]$	Refers to recipient address
$[\![\tilde{\mathbf{y}}$ contains "reply-to:"$]\!]$	Refers to a reply-to address
$[\![\tilde{\mathbf{y}}$ contains attachment$]\!]$	Expression refers to an attachment

Table 4: Input features that refer to properties of $\tilde{\mathbf{y}}$ for model $f_{\mathbf{v}}$.

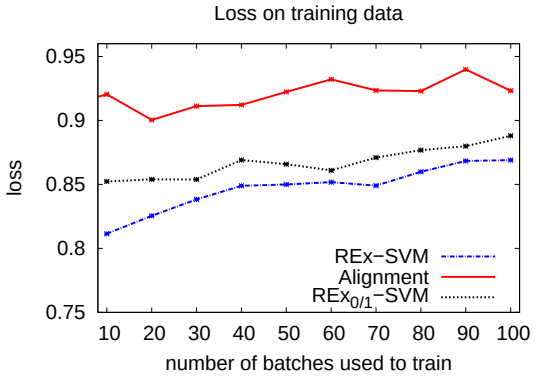


Figure 10: Average loss $\Delta_{\mathbf{u}}$ on training data for a varying number of training batches. Error bars indicate standard errors.

Figure 11 shows how the loss on the test data set changes when we varying the regularization parameters $C_{\mathbf{u}}$ and $C_{\mathbf{v}}$.

Feature	**Description**
Constant symbols in $\hat{\mathbf{y}}$	Number of constant symbols in the top-most concatenation
Non-constant subexpressions in $\hat{\mathbf{y}}$	Number of non-constant arguments of the top-most concatenation
$[\![\hat{\mathbf{y}}$ contains Latin characters$]\!]$	
$[\![\hat{\mathbf{y}}$ contains Greek characters$]\!]$	
$[\![\hat{\mathbf{y}}$ contains Russian characters$]\!]$	
$[\![\hat{\mathbf{y}}$ contains Asian characters$]\!]$	
$[\![\hat{\mathbf{y}}$ contains "subject:"$]\!]$	Expression refers to subject line
$[\![\hat{\mathbf{y}}$ contains "from:"$]\!]$	Expression contains a sender address
$[\![\hat{\mathbf{y}}$ contains "to:"$]\!]$	Contains a recipient address
$[\![\hat{\mathbf{y}}$ contains "reply-to:"$]\!]$	Contains a reply-to address
$[\![\hat{\mathbf{y}}$ contains attachment$]\!]$	Expression refers to attachment
$[\![\hat{\mathbf{y}}$ starts with "subject:" and ends with \n$]\!]$	Expression only refers to subject line
$[\![\hat{\mathbf{y}}$ starts with "from:" and ends with \n$]\!]$	Expression only refers to sender address
$[\![\hat{\mathbf{y}}$ starts with "to:" and ends with \n$]\!]$	Expression only refers to recipient address
$[\![\hat{\mathbf{y}}$ starts with "reply-to:" and ends with \n$]\!]$	Only refers to reply-to address
$[\![\hat{\mathbf{y}}$ starts with "attachment:" and ends with \n$]\!]$	Contains only refers to attachment
$[\![\hat{\mathbf{y}}$ starts with "subject:"$]\!]$	Expression starts with a subject line
$[\![\hat{\mathbf{y}}$ starts with "from:"$]\!]$	Starts with a sender address
$[\![\hat{\mathbf{y}}$ starts with "to:"$]\!]$	Starts with a recipient address
$[\![\hat{\mathbf{y}}$ starts with "reply-to:"$]\!]$	Starts with a reply-to address
$[\![\hat{\mathbf{y}}$ starts with "attachment:"$]\!]$	Starts with a subject line
$[\![\hat{\mathbf{y}}$ ends with "subject:"$]\!]$	Ends with a subject line
$[\![\hat{\mathbf{y}}$ ends with "from:"$]\!]$	Ends with a sender address
$[\![\hat{\mathbf{y}}$ ends with "to:"$]\!]$	Ends with a recipient address
$[\![\hat{\mathbf{y}}$ ends with "reply-to:"$]\!]$	Ends with a reply-to address
$[\![\hat{\mathbf{y}}$ ends with "attachment:"$]\!]$	Ends with reference to attachment
number of newlines in $\hat{\mathbf{y}}$	Number of line breaks in the expression
$[\![\hat{\mathbf{y}}$ contains a URL$]\!]$	
$[\![\hat{\mathbf{y}}$ is only a URL$]\!]$	
$[\![\hat{\mathbf{y}}$ contains an email address$]\!]$	
$[\![\hat{\mathbf{y}}$ is only an email address$]\!]$	
$[\![\hat{\mathbf{y}}$ contains a phone number$]\!]$	
$[\![\hat{\mathbf{y}}$ is only a phone number$]\!]$	
$[\![\hat{\mathbf{y}}$ contains an IP address$]\!]$	
$[\![\hat{\mathbf{y}}$ contains an attachment of type .exe$]\!]$	
$[\![\hat{\mathbf{y}}$ contains an attachment of type .jpg$]\!]$	
$[\![\hat{\mathbf{y}}$ contains an attachment of type .zip$]\!]$	
$[\![\hat{\mathbf{y}}$ contains an attachment of type .html$]\!]$	
$[\![\hat{\mathbf{y}}$ contains an attachment of type .doc$]\!]$	
$[\![\hat{\mathbf{y}}$ contains substring $\in S_{spam}]\!]$	Contains terms from the highest-scoring bag-of-words features for spam

Table 5: Output features that refer to properties of $\hat{\mathbf{y}}$ for model $f_{\mathbf{v}}$.

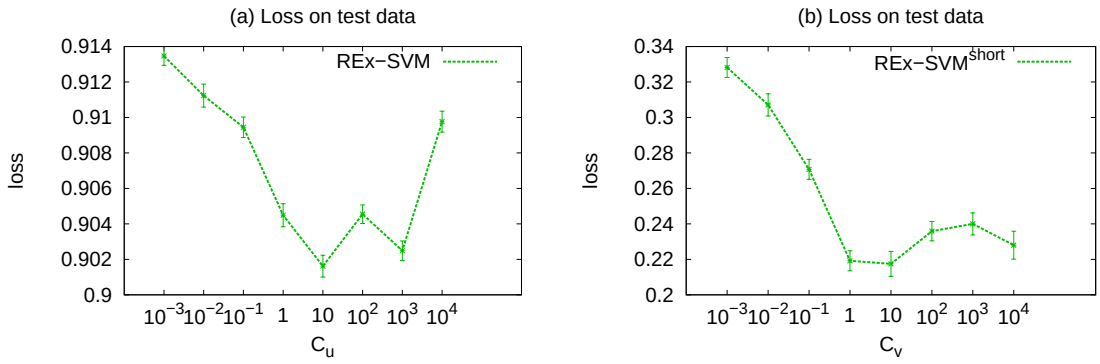


Figure 11: Average loss on test data for a varying regularization parameters $C_{\mathbf{u}}$ and $C_{\mathbf{v}}$ to train a model $\mathbf{f_u}$ (a) and a model $\mathbf{f_v}$, respectively. Error bars indicate standard errors.

A.4 Syntax of Postmasters' Regular Expressions

This section summarizes the syntactic constructs used by postmasters and their frequency. These observations provide the rationale behind the definition of the constrained search space of Algorithm 1. Table 6 shows the frequency at which macros occur in the ESP data set. Table 7 shows which iterators (*, $^+$, ?, $\{x\}$, $\{x,y\}$ for $x, y \in \mathbb{N}$) postmasters use as a suffix of the disjunction of characters (*e.g.,* [abc]* or [0-9]$^+$). Table 8 counts the frequency of iterators in conjunction with an alternative of regular expressions (*e.g.,* (girl|woman)?).

Macro	Frequency
\d	97
\S	71
\e	16
A-Z	25
a-z	86
A-F	28
a-f	17
0-9	65

Table 6: Macros used in the postmasters' expressions.

We measure the maximum nesting depth of alternatives of regular expression in the ESP data set: We find that 95.6% have a nesting depth of at most one—that is, they contain no layer of alternatives within the top-most alternative, such as a[a-z]$^+$. Only 4.4% have a greater nesting depth (e.g. a([a-z]$^+$|01), having a nesting depth of two). Algorithm 1 constructs the set of possible specializations of the j-th wildcard, starting with all subexpressions of all expressions in the training data. Hence, the nesting depth of alternatives

Iterator	Frequency
$[\ldots]$	21
$[\ldots]^*$	2
$[\ldots]^+$	73
$[\ldots]?$	0
$[\ldots]\{x\}$	49
$[\ldots]\{x,y\}$	39

Table 7: Iterators used in conjunction with a character disjunction—*e.g.,* [abc0-9]*.

Iterator	Frequency
$(\ldots\vert\ldots)$	166
$(\ldots\vert\ldots)^*$	0
$(\ldots\vert\ldots)^+$	0
$(\ldots\vert\ldots)?$	2
$(\ldots\vert\ldots)\{x\}$	0
$(\ldots\vert\ldots)\{x,y\}$	0

Table 8: Iterators used in conjunction with alternatives—*e.g.,* (viagra|cialis)$^+$.

in the constrained search space is at least the nesting depth of the training data. In line 6, the alternative of constant strings aligned at the j-th wildcard symbol is added; hence, the constrained search space has a nesting depth of at least one, even if the training data have a nesting depth of zero. For all character alternatives in the set of possible specializations, all macros from Table 6 and all iterators shown in Tables 7 and 8 are added.

Journal of Machine Learning Research 16 (2015) 3721-3756 Submitted 3/14; Revised 11/14; Published 12/15

Non-Asymptotic Analysis of a New Bandit Algorithm for Semi-Bounded Rewards

Junya Honda HONDA@STAT.T.U-TOKYO.AC.JP
Department of Complexity Science and Engineering
The University of Tokyo
Kashiwa-shi, Chiba, 277-8561, Japan

Akimichi Takemura TAKEMURA@STAT.T.U-TOKYO.AC.JP
Department of Mathematical Informatics
The University of Tokyo
Bunkyo-ku, Tokyo, 113-8561, Japan

Editor: Olivier Teytaud

Abstract

In this paper we consider a stochastic multiarmed bandit problem. It is known in this problem that Deterministic Minimum Empirical Divergence (DMED) policy achieves the asymptotic theoretical bound for the model where each reward distribution is supported in a known bounded interval, say $[0, 1]$. However, the regret bound of DMED is described in an asymptotic form and the performance in finite time has been unknown. We modify this policy and derive a finite-time regret bound for the new policy, Indexed Minimum Empirical Divergence (IMED), by refining large deviation probabilities to a simple non-asymptotic form. Further, the refined analysis reveals that the finite-time regret bound is valid even in the case that the reward is not bounded from below. Therefore, our finite-time result applies to the case that the minimum reward (that is, the maximum loss) is unknown or unbounded. We also present some simulation results which shows that IMED much improves DMED and performs competitively to other state-of-the-art policies.

Keywords: stochastic bandit, finite-time regret, large deviation principle

1. Introduction

In the multiarmed bandit problem a gambler pulls arms of a slot machine sequentially so that the total reward is maximized. There is a tradeoff between exploration and exploitation since he cannot know the most profitable arm unless pulling all arms infinitely many times.

There are two main formulations for this problem: stochastic and nonstochastic bandits. In the stochastic setting rewards of each arm follow an unknown distribution (Agrawal, 1995; Gittins, 1989; Vermorel and Mohri, 2005) whereas the rewards are determined by an adversary in the nonstochastic setting (Auer et al., 2002b). In this paper we consider the K-armed stochastic bandit, where rewards of arm $i \in \{1, 2, \cdots, K\}$ are i.i.d. sequence from unknown distribution $F_i \in \mathcal{F}$ with expectation μ_i for a model $\mathcal{F}$ known to the gambler. For the maximum expectation $\mu^* \equiv \max_i \mu_i$, we call an arm i optimal if $\mu_i = \mu^*$ and suboptimal otherwise. If the gambler knows each μ_i beforehand, it is best to choose optimal arms at every round. A *policy* is a strategy of the gambler for choosing arms based on the past results

of plays. The performance of a policy is usually measured by *pseudo-regret*, or simply *regret* in short. This is the gap of cumulative expectations between the optimal choice and the actual choice, which is expressed as

$$\mathcal{R}(n) \equiv \sum_{i:\mu_i<\mu^*} (\mu^* - \mu_i) T_i(n)\,,$$

where $T_i(n)$ is the number of plays of arm i through the first n rounds.

1.1 Theoretical Bound and its Achievability

Robbins (1952) first considered this setting and Lai and Robbins (1985) gave a framework for determining an optimal policy by establishing an asymptotic theoretical bound for the regret. Later this theoretical bound was extended to multiparameter or nonparametric models $\mathcal{F}$ by Burnetas and Katehakis (1996). It is proved in their paper that under a mild regularity condition any policy satisfies

$$\mathrm{E}[T_i(n)] \geq \frac{\log n}{D_{\inf}(F_i, \mu^*; \mathcal{F})} - \mathrm{o}(\log n) \tag{1}$$

for any suboptimal arm i, where $D_{\inf}(F, \mu; \mathcal{F})$ is defined in terms of Kullback-Leibler divergence $D(\cdot\|\cdot)$ by

$$D_{\inf}(F, \mu; \mathcal{F}) = \inf_{G\in\mathcal{F}:\mathrm{E}_G[X]>\mu} D(F\|G)\,.$$

The most popular model in the nonparametric setting is the family of distributions with supports contained in a known bounded interval, say $[0, 1]$. For this model, which we denote by $\mathcal{A}_0$, it is known that fine performance can be obtained by policies called Upper Confidence Bound (UCB) (Auer et al., 2002a; Audibert et al., 2009; Cappé et al., 2013). However, although some bounds for regrets of UCB policies have been obtained in a non-asymptotic form, they do not necessarily achieve the asymptotic theoretical bound.

Recently Honda and Takemura (2010) proposed Deterministic Minimum Empirical Divergence (DMED) policy, which chooses arms based on the value of $D_{\inf}(\hat{F}_i, \mu; \mathcal{A}_0)$, or simply written as $D_{\inf}(\hat{F}_i, \mu)$, for empirical distribution $\hat{F}_i$ of arm i. Whereas DMED achieves the asymptotic theoretical bound, the evaluation heavily depends on an asymptotic analysis and any finite-time regret bound has been unknown.

In this paper, we consider the family $\mathcal{A}$ of distributions on $(-\infty, 1]$ instead of the bounded support model $\mathcal{A}_0$. We first show that $D_{\inf}(F, \mu; \mathcal{A}_0) = D_{\inf}(F, \mu; \mathcal{A})$ for all $F \in \mathcal{A}_0$. Thus, any asymptotically optimal policy for the model $\mathcal{A}$ is also asymptotically optimal for $\mathcal{A}_0$, even though the gambler has more candidates for the true distribution of each arm in the model $\mathcal{A}$ than in $\mathcal{A}_0$.

We next propose a policy, the *IMED (Indexed Minimum Empirical Divergence)* algorithm. This is an indexed version of DMED in the sense that IMED simply chooses an arm which minimizes an index at each round whereas DMED requires to keep a list of arms to be pulled. We derive a finite-time regret bound of IMED for any distribution in $\mathcal{A}$ such that moment generating function $\mathrm{E}[e^{\lambda X}]$ exists in some neighborhood of $\lambda = 0$. The derived bound coincides with the asymptotic theoretical bound and therefore IMED is

asymptotically optimal for both $\mathcal{A}$ and $\mathcal{A}_0$. Since nonstochastic bandits inevitably require the boundedness of the support, we see that an advantage of assuming stochastic bandits is that the semi-bounded rewards can be dealt with in this nonparametric setting. Furthermore, we show that the reminder term of the logarithmic regret of IMED is $\mathrm{O}(1)$, whereas they are $\mathrm{O}((\log n)^a)$, $0 < a < 1$, in previously known asymptotically optimal regret bounds.

Note that DMED policy can be implemented without knowledge of the lower bound of the reward and achieves the asymptotic bound if the reward is only bounded from below by some unknown value. In this sense it is intuitively not surprising that DMED or its variant achieves the asymptotic the semi-bounded reward. However, the theoretical analysis for DMED in Honda and Takemura (2010) heavily depends on the boundedness of the support and its extension is not theoretically obvious.

There has also been some research for the nonparametric stochastic bandit with *unbounded* support distributions (Bubeck et al., 2012; Liu and Zhao, 2011). In particular, it is shown in Bubeck et al. (2012) that a logarithmic regret can be achieved if, for some $\epsilon > 0$, $\mathrm{E}_{F_i}[|X|^{1+\epsilon}]$ is bounded by a value *known to the gambler* beforehand. Although our assumption of the existence of the moment generating function $\mathrm{E}_{F_i}[\mathrm{e}^{\lambda X}]$ is more restrictive than the existence of the moment $\mathrm{E}_{F_i}[|X|^{1+\epsilon}]$, IMED does not require any knowledge on the value of $\mathrm{E}_{F_i}[\mathrm{e}^{\lambda X}]$ (or $\mathrm{E}_{F_i}[|X|^{1+\epsilon}]$). Therefore our assumption is not comparable to that in Bubeck et al. (2012).

1.2 Motivation for Semi-bounded Support Model

An example such that the lower bound of the reward is unknown or unbounded is the minimization of the sum of the time-delays in some task such as network routing (Vermorel and Mohri, 2005; Krishnamurthy et al., 2001), where the agent has many sources to obtain the same data. In this case, it may take a long time to complete the task and it is natural to consider that the reward (that is, negative of the time-delay) is not bounded from below. One may wonder that if some time-limit is fixed then the problem becomes a bounded bandit and a good finite-time regret has been already achieved by, for example, kl-UCB in Cappé et al. (2013) (although the regret bound of kl-UCB is not asymptotically optimal for distributions other than Bernoulli distributions). However, the time-limit (or the maximum time-delay) is usually set "conservatively", that is, set to a value much larger than time-delays in usual tries. In such a case, policies based only on empirical means tend to work poorly (see also Audibert et al., 2009). For example, kl-UCB achieves a regret near

$$\sum_{i:\mu_i<\mu^*} \frac{\mu^* - \mu_i}{D(\mathrm{B}(\mu_i)\|\mathrm{B}(\mu^*))} \log n$$

for reward distributions on $[0,1]$, where $\mathrm{B}(\mu)$ denotes the Bernoulli distribution with mean μ. On the other hand, if the gambler conservatively estimates the lower bound of the reward by $a < 0$ instead of 0, he applies the policy after the rescaling from $[a,1]$ to $[0,1]$ and the regret becomes

$$\sum_{i:\mu_i<\mu^*} \frac{\mu^* - \mu_i}{D(\mathrm{B}((\mu_i - a)/(1-a))\|\mathrm{B}((\mu^* - a)/(1-a)))} \log n\,,$$

which goes to infinity as $a \to -\infty$. Audibert et al. (2009) overcame this problem by UCB-V policy, which uses empirical variances as well as empirical means. However, in turn, UCB-V does not necessarily perform well for usual Bernoulli distributions as reported in Cappé et al. (2013). Therefore the IMED policy has an advantage since it always achieves the optimal regret bound, which does not depend on whether the gambler knows the lower bound of the reward or not.

1.3 Outline

This paper is organized as follows. In Sect. 2 we give definitions used throughout this paper and propose the IMED policy as an indexed version of DMED. In Sect. 3, we give the main results of this paper on the finite-time regret bound of IMED for distributions on $(-\infty, 1]$. We discuss relation between IMED and other policies in Sect. 4 and give some simulation results of these policies in Sect. 5. The remaining sections and appendices are devoted to the proof of the main theorems. In Sect. 6, we analyze properties of the function $D_{\inf}$ for our model. In Sect. 7, we derive a large deviation probability of an empirical distribution $\hat{F}_t$ measured with $D_{\inf}$ in a non-asymptotic form. By using this probability, we derive the finite-time regret bound of IMED in Sect. 8. We conclude this paper with some discussion on the regularity condition assumed throughout the paper in Sect. 9. We evaluate constants used in the finite-time regret bound in Appendix A. We give a proof of a lemma analogous to the bounded-support model in Appendix B. Finally we prove the asymptotic but refined regret bound of IMED in Appendix C.

2. Preliminaries

In this section we introduce notation used throughout this paper and propose the IMED policy.

2.1 Notation

Let $\mathcal{A}_a$, $a \in (-\infty, 1)$, be the family of probability distributions on $[a, 1]$. We denote the family of distributions on $(-\infty, 1]$ by $\mathcal{A}_{-\infty}$ or simply $\mathcal{A}$. For $F \in \mathcal{A}$, the cumulative distribution at a point $x \in \mathbb{R}$ is denoted by $\bar{F}(x) \equiv F((-\infty, x])$, where $F(A)$, $A \subset \mathbb{R}$, denotes the measure of a set A. $\mathrm{E}_F[\cdot]$ denotes the expectation under $F \in \mathcal{A}$. When we write, for example, $\mathrm{E}_F[u(X)]$ for a function $u : \mathbb{R} \to \mathbb{R}$, X denotes a random variable with distribution F. The expectation of F is denoted by $\mathrm{E}(F) \equiv \mathrm{E}_F[X]$.

Let $J(n) \in \{1, 2, \cdots, K\}$ be the arm pulled at the n-th round. We define $T_i(n)$ as the number of times that arm i has been pulled through the first n rounds. Then, we have $T_i(n) = \sum_{l=1}^{n} \mathbb{1}[J(l) = i]$ where $\mathbb{1}[\cdot]$ denotes the indicator function. $\hat{F}_{i,t}$ and $\hat{\mu}_{i,t}$ denote the empirical distribution and the mean of arm i when arm i is pulled t times. $\hat{F}_i(n) \equiv \hat{F}_{i,T_i(n)}$ and $\hat{\mu}_i(n) \equiv \hat{\mu}_{i,T_i(n)}$ denote the empirical distribution and the mean of arm i at the n-th round. The largest empirical mean after the first n rounds is denoted by $\hat{\mu}^*(n) \equiv \max_i \hat{\mu}_i(n)$.

The function $D_{\inf}$ defined as

$$D_{\inf}(F, \mu; \mathcal{A}_a) \equiv \inf_{G \in \mathcal{A}_a : \mathrm{E}(G) > \mu} D(F \| G)$$

Algorithm 1 IMED Policy

Initialization: Pull each arm once.

Loop: Choose an arm i minimizing

$$I_i(n) \equiv T_i(n) D_{\inf}(\hat{F}_i(n), \hat{\mu}^*(n); \mathcal{A}) + \log T_i(n)\,,$$

where the tie-breaking rule is arbitrary.

plays a central role in the DMED policy in Honda and Takemura (2010) and the IMED policy defined below. Let

$$\begin{aligned} L(\nu; F, \mu) &\equiv \mathrm{E}_F[\log(1 - (X - \mu)\nu)]\,, \\ L_{\max}(F, \mu) &\equiv \max_{0 \le \nu \le \frac{1}{1-\mu}} L(\nu; F, \mu)\,. \end{aligned} \tag{2}$$

Functions L and $L_{\max}$ correspond to the Lagrangian function and the dual problem of $D_{\inf}(F, \mu; \mathcal{A})$, respectively. The following proposition shows that $D_{\inf}$ is equal to $L_{\max}$ in the case of the bounded support model $\mathcal{A}_0$. In Sect. 3 we prove that the same result holds for the semi-bounded support model $\mathcal{A}$.

Proposition 1 (Honda and Takemura, 2010, Theorem 5) *For all $F \in \mathcal{A}_0$ and $\mu < 1$ it holds that $D_{\inf}(F, \mu; \mathcal{A}_0) = L_{\max}(F, \mu)$.*

2.2 IMED Policy

In the model $\mathcal{A}_0$, Honda and Takemura (2010) proposed an asymptotically optimal policy, DMED, which maintains the list of arms satisfying

$$T_i(n) D_{\inf}(\hat{F}_i(n), \hat{\mu}^*(n); \mathcal{A}_0) + \log T_i(n) \le \log n \tag{3}$$

where The DMED policy pulls an arm from the list in some order.

In this paper, we use the left-hand side of (3) as the index $I_i(n)$ for choosing an arm. Our proposed policy, Indexed Minimum Empirical Divergence (IMED) policy, is described as Algorithm 1. In the index $I_i(n)$, the first term $T_i(n) D_{\inf}(\hat{F}_i(n), \hat{\mu}^*(n)) \ge 0$ corresponds to the penalty for empirical distributions unlikely to occur from a distribution with expectation larger than $\hat{\mu}^*(n)$ and IMED usually chooses a currently optimal arm i since it satisfies $D_{\inf}(\hat{F}_i(n), \hat{\mu}^*(n)) = 0$. The second term $\log T_i(n)$ is the penalty for arms pulled too many times and corresponds to the exploration function.

Note that here we say that IMED is an index policy in a weaker sense than other index policies. Although both IMED and well known index policies such as Gittins index (Gittins, 1989) and UCB choose an arm which maximizes or minimizes its index at each round, the values of Gittins index and UCB score of each arm can be determined only from samples of the corresponding arm. On the other hand, the index of IMED also requires the maximum empirical mean over all arms, which depends on statistics of other arms. It may seem somewhat unnatural to use such an index for choosing an arm but IMED has an advantage in the computational complexity for this property of the index as discussed in Sect. 4.1.

3. Main Results

We now state the main results of this paper in Theorems 2, 3 and 5. In Theorem 2, we show that the theoretical bound does not depend on knowledge of the lower bound of the support. In Theorem 3, we give a non-asymptotic regret bound of IMED, which shows that the theoretical bound can be achieved by IMED. We give an asymptotic but refined regret bound of IMED in Theorem 5.

Theorem 2 *Let $a \in [-\infty, 1)$ and $F \in \mathcal{A}_a$ be arbitrary.* (i) $D_{\inf}(F, \mu; \mathcal{A}_a) = D_{\inf}(F, \mu; \mathcal{A})$. (ii) *If $\mu < 1$ then*

$$D_{\inf}(F, \mu; \mathcal{A}) = L_{\max}(F, \mu)\,.$$

We prove this theorem in Sect. 6. The part (i) of this theorem means that the theoretical bound does not depend on whether the gambler knows lower bound of the support of distributions or he has to consider the case that the support is not bounded from below. Furthermore, from (ii), we can compute $D_{\inf}(F, \mu; \mathcal{A})$ by using the expression $L_{\max}(F, \mu)$ as in the case of $\mathcal{A}_0$. In view of this theorem we sometimes write $D_{\inf}(F, \mu)$ instead of more precise $D_{\inf}(F, \mu; \mathcal{A}_a)$ or $D_{\inf}(F, \mu; \mathcal{A})$.

Define

$$\nu_i^* \equiv \operatorname*{argmax}_{0 \le \nu \le \frac{1}{1-\mu^*}} \mathrm{E}_{F_i}[\log(1 - (X - \mu^*)\nu)]\,,$$

$$\lambda_{i,\mu} \equiv \sup\left\{\lambda \in \mathbb{R} \cup \{\infty\} : \mathrm{E}_{F_i}\left[\left(\frac{1-X}{1-\mu}\right)^{\lambda}\right] \le 1\right\}, \tag{4}$$

where we show that ν_i^* exists uniquely when $\mathrm{E}(F_i) < \mu^*$ in Sect. 6 and show $\lambda_{i,\mu} > 1$ for $\mu < \mu_i$ in Sect. 7. We further define Fenchel-Legendre transforms of cumulant generating functions of random variables X and $\log(1 - (X - \mu^*)\nu_i^*)$ as

$$\Lambda_i^*(x) \equiv \sup_{\lambda}\{\lambda x - \log \mathrm{E}_{F_i}[\mathrm{e}^{\lambda X}]\}\,, \tag{5}$$

$$\tilde{\Lambda}_i^*(x) \equiv \sup_{\lambda}\left\{\lambda x - \log \mathrm{E}_{F_i}[(1 - (X - \mu^*)\nu_i^*)^{\lambda}]\right\}. \tag{6}$$

Then, for[1] $\Delta_i \equiv \mu^* - \mu_i$ and $\mathcal{I}_{\mathrm{opt}} \equiv \{j : \mu_j = \mu^*\} \subset \{1, \cdots, K\}$, the regret of IMED is bounded as follows.

Theorem 3 *Assume that $\mu^* < 1$ and $\mathrm{E}_{F_j}[\mathrm{e}^{\lambda X}] < \infty$ in some neighborhood of $\lambda = 0$ for some $j \in \mathcal{I}_{\mathrm{opt}}$. Then, for any fixed $0 < \delta < \min_{i:\mu_i<\mu^*} \Delta_i/2$, the expected number of pulls of a suboptimal arm $i \notin \mathcal{I}_{\mathrm{opt}}$ is bounded as*

$$\mathrm{E}[T_i(n)] \le \frac{\log n}{D_{\inf}(F_i, \mu^*) - \frac{2\delta}{1-\mu^*}} + \frac{1}{1 - \mathrm{e}^{-\tilde{\Lambda}_i^*(D_{\inf}(F_i,\mu^*) - \frac{\delta}{1-\mu^*}))}} + \min_{j \in \mathcal{I}_{\mathrm{opt}}}\left\{\frac{6\mathrm{e}}{(1 - 1/\lambda_{j,\mu^*-\delta})(1 - \mathrm{e}^{-(1-1/\lambda_{j,\mu^*-\delta})\Lambda_j^*(\mu^*-\delta)})^3}\right\}.$$

1. We often use the subscript i for a suboptimal arm and use j for an optimal arm.

Consequently, the expected regret is bounded as

$$\mathrm{E}[\mathcal{R}(n)] \le \sum_{i:\Delta_i>0} \Delta_i \left(\frac{\log n}{D_{\inf}(F_i,\mu^*) - \frac{2\delta}{1-\mu^*}} + \frac{1}{1-\mathrm{e}^{-\tilde{\Lambda}_i^*(D_{\inf}(F_i,\mu^*)-\frac{\delta}{1-\mu^*}))}} \right) + \left(\sum_{i=1}^K \Delta_i\right) \min_{j\in\mathcal{I}_{\mathrm{opt}}} \left\{ \frac{6\mathrm{e}}{(1-1/\lambda_{j,\mu^*-\delta})(1-\mathrm{e}^{-(1-1/\lambda_{j,\mu^*-\delta})\Lambda_j^*(\mu^*-\delta)})^3} \right\}.$$

We prove Theorem 3 in Sect. 8 based on non-asymptotic large deviation probabilities for $D_{\inf}(\hat{F}_i(n), \hat{\mu}^*(n))$ given in Sect. 7. In Appendix A, we discuss simple representations of $(\lambda_{j,\mu},\ \Lambda_i^*(x),\ \tilde{\Lambda}_i^*(x))$ and show that $\lambda_{j,\mu^*-\delta} = 1+\mathrm{O}(\delta)$, $\Lambda_i^*(\mu^*-\delta) = \mathrm{O}(\delta^2)$ and $\tilde{\Lambda}_i^*(D_{\inf}(F_i,\mu^*) - \delta/(1-\mu^*)) = \mathrm{O}(\delta^2)$. The following corollary is straightforward from this observation.

Corollary 4 *Under the assumption of Theorem 3,*

$$\mathrm{E}[\mathcal{R}(n)] = \sum_{i:\mu_i<\mu^*} \frac{\Delta_i \log n}{D_{\inf}(F_i,\mu^*)} + \mathrm{O}((\log n)^{10/11}). \tag{7}$$

Proof From $1-\mathrm{e}^{-\epsilon} = \mathrm{O}(\epsilon)$ and the above observation on $(\lambda_{j,\mu},\ \Lambda_i^*(x),\ \tilde{\Lambda}_i^*(x))$,

$$\mathrm{E}[\mathcal{R}(n)] \le \sum_{i:\mu_i<\mu^*} \frac{\Delta_i \log n}{D_{\inf}(F_i,\mu^*)} + \mathrm{O}(\delta \log n) + \mathrm{O}(\delta^{-2}) + \mathrm{O}(\delta^{-10}).$$

We obtain (7) by letting $\delta = \mathrm{O}((\log n)^{-1/11})$. ■

From this corollary we see that IMED is asymptotically optimal in view of (1). However, the reminder term $\mathrm{O}((\log n)^{10/11})$ is quite larger than those of known asymptotically optimal policies for other models although our model, the semi-bounded support model, is quite complicated. For example, it is shown in Cappé et al. (2013) that the KL-UCB policy achieves the asymptotic bound with reminder term $\mathrm{O}(\sqrt{\log n})$ for a subclass of one-dimensional exponential families and $\mathrm{O}((\log n)^{4/5}\log\log n)$ for the finite support model. The following theorem shows that the reminder term can be much improved in our model.

Theorem 5 *(i) Assume that $\mu^* < 1$ and $\mathrm{E}_{F_i}[\mathrm{e}^{\lambda X}] < \infty$ in some neighborhood of $\lambda = 0$ for all $i \in \{1, 2, \cdots, K\}$. Then*

$$\mathrm{E}[\mathcal{R}(n)] = \sum_{i:\mu_i<\mu^*} \frac{\Delta_i \log n}{D_{\inf}(F_i,\mu^*)} + \mathrm{O}(1). \tag{8}$$

(ii) Furthermore, if the distribution of each arm has a bounded support then the reminder term $\mathrm{O}(1)$ in (8) can be replaced with $-\mathrm{O}(\log\log n)$, that is, there exists $C > 0$ such that for all sufficiently large n

$$\mathrm{E}\left[\mathcal{R}(n)\right] \le \sum_{i:\mu_i<\mu^*} \frac{\Delta_i \log n}{D_{\inf}(F_i,\mu^*)} - C\log\log n. \tag{9}$$

The proof of this theorem is much more complicated than that of Theorem 3 and given in Appendix C.

Note that a policy asymptotically optimal for the semi-bounded support model is also asymptotically optimal for the model of finite-support distributions (Honda and Takemura, 2011, Theorem 3). Therefore the regret bound (9) of IMED is asymptotically better than that of KL-UCB in Cappé et al. (2013) for finite-support distributions, of which the reminder term is $\mathrm{O}((\log n)^{4/5} \log\log n)$.

To the best of the authors' knowledge, this is the first result to show that the asymptotic bound (1) is achievable with a reminder term $\mathrm{O}(1)$ instead of $\mathrm{o}(\log n)$. The key to this refined bound is to apply a technique for a stopping-time of a stochastic process, which we evaluate in Lemma 18. The authors think that the regret bounds of other policies can also be improved by using this novel technique.

4. Relation with Other Policies

In the previous sections we showed that IMED achieves the asymptotic bound for the semi-bounded support model. In this section we compare IMED with other policies which achieve a logarithmic regret for some models.

4.1 KL-UCB Policies

Burnetas and Katehakis (1996) proposed a UCB policy for a general class $\mathcal{F}$ which chooses an arm maximizing the index

$$\sup\left\{\mu : T_i(n) D_{\inf}(\hat{F}_i(n), \mu; \mathcal{F}) \le f(n)\right\} \tag{10}$$

for some exploration function $f(n)$. They gave a sufficient condition for the asymptotic optimality of this policy for general model $\mathcal{F}$ and proved that the condition is satisfied for the finite support model and the normal distribution model with known variances. Furthermore Cappé et al. (2013) proved its asymptotic optimality with a finite-time regret bound for the finite support model and a subclass of exponential families. They also proved that this policy where $D_{\inf}(\mu; \mathcal{F})$ is replaced with the Bernoulli divergence

$$D_{\inf}(\hat{F}_i(n), \mu; \mathcal{F}_{\mathrm{Ber}}) = \hat{\mu}_i(n) \log \frac{\hat{\mu}_i(n)}{\mu} + (1 - \hat{\mu}_i(n)) \log \frac{1 - \hat{\mu}_i(n)}{1 - \mu} \tag{11}$$

achieves a logarithmic regret for general distributions with supports in $[0, 1]$. We refer to this policy for general model $\mathcal{F}$ as KL-UCB and the policy with (11) for bounded-support distributions as kl-UCB after Cappé et al. (2013). We can make the KL-UCB policy computationally feasible by using Prop. 1 and Theorem 2 for the bounded support model and the semi-bounded support model, respectively, but the asymptotic optimality for these models has been currently unknown although the authors believe that it can be proved as in IMED by using Theorem 2 and large deviation probabilities evaluated in the next section.

Other than the theoretical guarantee of the asymptotic optimality, the IMED has an advantage in the computational complexity. In the semi-bounded support model (or the bounded support model), the computation of $D_{\inf}$ itself involves an optimization and a

simple representation of (10) has not been known whereas $D_{\inf}$ can be represented as a *univariate* convex optimization as shown in Theorem 2.

Furthermore, since $D_{\inf}(F, \mu) = 0$ for $\mathrm{E}(F) = \mu$, IMED does not require the computation of $D_{\inf}(\hat{F}_i(n), \hat{\mu}^*(n))$ for currently optimal arms and the computation of these values for currently suboptimal arms are sufficient. Since any suboptimal arm is pulled at most $\mathrm{O}(\log n)$ times in average, the size of the support of $\hat{F}_i(n)$ is $\mathrm{O}(\log n)$ and the average complexity of IMED at each round becomes $\mathrm{O}(\log n)$. On the other hand, KL-UCB also require the computation of (10) for currently optimal arms and the complexity becomes $\mathrm{O}(n)$ as discussed in Cappé et al. (2013, Sect. 6.2). This advantage of IMED justifies to some extent the use of a somewhat unnatural index which depends on statistics of other arms.

4.2 Bayesian Policies

There have also been some Bayesian policies which are known to achieve the asymptotic bound for some model.

The Bayes-UCB policy (Kaufmann et al., 2012a) is a variant of UCB family obtained by the replacement of $T_i(n)D_{\inf}(\hat{F}_i(n), \mu)$ in (10) with a quantity associated with a posterior probability on the true expectation of the arm. The asymptotic optimality of this policy is proved for the Bernoulli model.

Another Bayesian policy is Thompson sampling (TS) originally proposed in Thompson (1933), which is a randomized algorithm which chooses an arm according to the posterior probability that the arm is optimal. TS is proved to be asymptotically optimal for general one-dimensional exponential families including the Bernoulli model (Kaufmann et al., 2012b; Agrawal and Goyal, 2013; Korda et al., 2013). It is also reported that TS is easily applicable to many models with a state-of-the-art performance (Chapelle and Li, 2012; Russo and Roy, 2013). On the other hand, TS requires random sampling from the posterior which is difficult for models other than exponential families, particularly in the nonparametric models. Although it may become tractable for the semi-bounded support model in non-parametric Bayesian framework, it is not very simple compared to the computation of $D_{\inf}$ and it remains unknown whether TS works practically for our model.

4.3 Achievability of Logarithmic Regret for Semi-bounded Support Model

Another question is whether or not there exists a simpler policy than IMED which achieves a (possibly non-optimal) logarithmic regret for the semi-bounded support model. For the bounded support model a logarithmic regret can be achieved by kl-UCB policy as described above. The key property of KL-UCB is

$$D(\mathrm{B}(\mathrm{E}(F))\|\mathrm{B}(\mu)) \leq D_{\inf}(F, \mu)$$

for $F \in \mathcal{A}_0$, which means that the Bernoulli divergence can be used as a lower bound of $D_{\inf}(F, \mu)$ when the expectation (that is, the first-order moment) of F is specified. However, in the derivation of this inequality a convex function on the support $[0, 1]$ is bounded from *above* and the lower and upper bounds of the support are explicitly required (see Sect. 6.1 of Cappé et al. (2013) for detail), which makes difficult to bound $D_{\inf}(F, \mu)$ for general $F \in \mathcal{A}$.

A natural idea to bound $D_{\inf}(F,\mu)$ is to use higher-order moments of F. DMED-M (Honda and Takemura, 2012) is a policy based on this idea and obtained by replacing $D_{\inf}(F,\mu) = D_{\inf}(F,\mu;\mathcal{A}_a)$ for $F \in \mathcal{A}_a$, $a > -\infty$, with

$$D^{(d)}_{\inf}(\boldsymbol{M}^{(d)},\mu;\mathcal{A}_a) \equiv \inf_{G\in\mathcal{A}_a:\mathrm{E}_G[X^m]=\mathrm{E}_F[X^m],\, m=1,2,\cdots,d} D_{\inf}(G,\mu;\mathcal{A}_a)\,,$$

where $\boldsymbol{M}^{(d)} = (\mathrm{E}_F[X], \mathrm{E}_F[X^2], \cdots, \mathrm{E}_F[X^d])$. We can compute $D^{(d)}_{\inf}$ by solving algebraic equations and it is expressed in an explicit form for $d \le 4$ from the theory of Tchebycheff system (Karlin and Studden, 1966). The important point is that $D^{(d)}_{\inf}$ for even d does not depend on the lower bound a of the support (Honda and Takemura, 2012, Theorem 3). This means that DMED-M for even d achieves a logarithmic regret bound without knowledge on the lower bound a of the support whereas a policy using Bernoulli divergence $D^{(1)}_{\inf}(\mathrm{E}(F),\mu;\mathcal{A}_a)$ becomes meaningless for $a \to -\infty$ as discussed in Introduction. Therefore we can expect that DMED-M, or other policies based on $D^{(d)}_{\inf}$, also achieves a logarithmic regret for the semi-bounded support model since the key technique, Tchebycheff system, is extended to semi-bounded support distributions (Karlin and Studden, 1966, Chap. V).

5. Experiment

In this section we give some simulation results for IMED, DMED, Thompson sampling (TS) and KL-UCB family. For the KL-UCB family, we use $f(n) = \log n$ as an exploration function for (10) since the asymptotic optimality is shown in Burnetas and Katehakis (1996) for some models and it is empirically recommended in Cappé et al. (2013) although the latter paper uses $f(n) = \log n + c \log\log n$ for some $c > 0$ in the proof of the optimality. The kl-UCB+ and KL-UCB+ (Garivier and Cappé, 2011) are empirical improvements of kl-UCB and KL-UCB, respectively, where $f(n) = \log n$ is replaced with $f(n) = \log(n/T_i(n))$. The optimality analysis of these policies has not been given but a similar version is discussed in Kaufmann (2014, Proposition 2.4) for some models.

Each plot is an average over 10,000 trials. In the four figures given below, IMED and KL-UCB+ performed almost the best. Whereas the complexity of IMED is smaller than KL-UCB family as discussed in Sect. 4.1, the regret of IMED was slightly worse than that of KL-UCB+.

First, Fig. 1 shows simulation results of IMED, DMED, TS, kl-UCB and kl-UCB+ for ten-armed bandit with Bernoulli rewards with $\mu_1 = 0.1$, $\mu_2 = \mu_3 = \mu_4 = 0.05$, $\mu_5 = \mu_6 = \mu_7 = 0.02, \mu_8 = \mu_9 = \mu_{10} = 0.01$, which is the same setting as those in[2] Kaufmann et al. (2012b) and Cappé et al. (2013).

Next, we consider the case that the time-delay X'_i for some task by the i-th agent follows an exponential distribution with density $\mathrm{e}^{-x/\mu'_i}/\mu'_i$, $x \ge 0$, and the player tries to minimize the cumulative delay. Since we modeled the reward as a random variable in $(-\infty, 1]$, we set

2. The simulation result for DMED in this paper is different from those in these references where DMED is reported to perform much worse. This is because a policy where (3) is replaced with the condition

$$T_i(n) D_{\inf}(\hat{F}_i(n), \hat{\mu}^*(n); \mathcal{A}_0) \le \log n$$

is used as "DMED" in these references although the optimality proof of DMED is given for (3). This replacement can be interpreted as that of KL-UCB+ with KL-UCB (see also Garivier and Cappé (2011)).

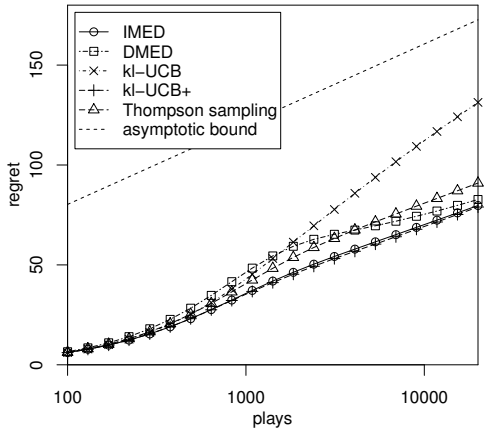


Figure 1: Average regret for 10-armed Bernoulli bandit.

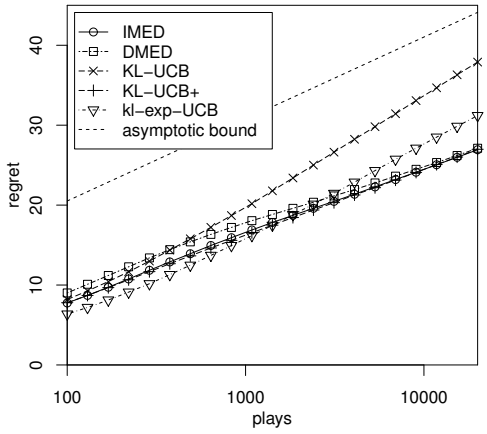


Figure 2: Average regret for 5-armed bandit where the negative reward follows an exponential distribution.

the reward as $X_i = 1 - X_i'$, that is, X_i has density $\mathrm{e}^{-(1-x)/\mu_i'}/\mu_i' = \mathrm{e}^{-(1-x)/(1-\mu_i)}/(1-\mu_i)$, $x \le 1$, with expectation $\mu_i = 1 - \mu_i'$. Fig. 2 shows simulation results for 5-armed bandit with $\mu_i' = 1/5, 1/4, 1/3, 1/2, 1$, that is, $\mu_i = 4/5, 3/4, 2/3, 1/2, 0$. We used IMED, DMED, KL-UCB, KL-UCB+ for $\mathcal{A}$ and KL-UCB for the (shifted) exponential distributions, which we refer as kl-exp-UCB, where the KL divergence is written as

$$D(\hat{\mu}_i \| \mu) = \frac{1-\hat{\mu}_i}{1-\mu} - 1 - \log \frac{1-\hat{\mu}_i}{1-\mu}\,.$$

The kl-exp-UCB policy explicitly assumes the knowledge that $1 - X_i$ follows an exponential distribution (and under the same assumption TS can also be implemented) whereas the other policies only uses the knowledge on the upper bound of the reward.

Since kl-exp-UCB is asymptotically optimal for exponential distributions, it is theoretically assured that it asymptotically outperforms other policies for this setting. Nevertheless, it seems from the comparison of kl-exp-UCB and KL-UCB that the gap between theoretical bounds for semi-bounded support model and for exponential distributions is not very large, which supports the effectiveness of the nonparametric model.

Finally, Figs. 3 and 4 show results of IMED, DMED, KL-UCB and KL-UCB+ for truncated normal distributions on $[0, 1]$ and $(-\infty, 1]$, respectively, as examples of multiparameter models. The cumulative distribution of each reward is given by

$$\bar{F}_i(x) = \begin{cases} 0, & x < a, \\ \frac{\Phi((x-\mu_i')/\sigma_i) - \Phi((a-\mu_i')/\sigma_i)}{\Phi((1-\mu_i')/\sigma_i) - \Phi((a-\mu_i')/\sigma_i)}, & a \le x < 1, \\ 1, & 1 \le x, \end{cases}$$

where $a = 0$ or $-\infty$, and Φ is the cumulative distribution function of the standard normal distribution. We also give results of kl-UCB and TS for the Bernoulli bandit for the setting

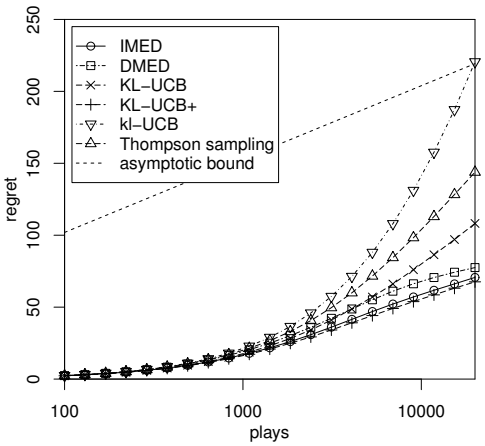


Figure 3: Average regret for 5-armed bandit with truncated normal distributions on $[0, 1]$.

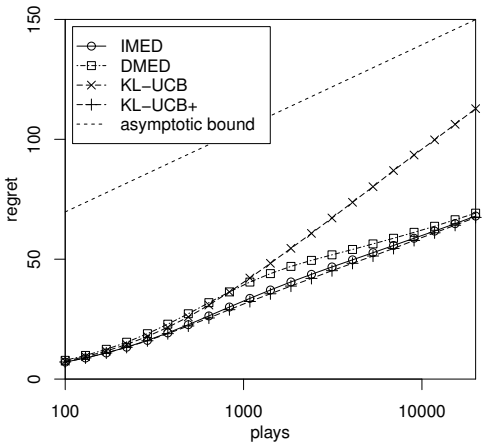


Figure 4: Average regret for 5-armed bandit with truncated normal distributions on $(-\infty, 1]$.

of Fig. 3 where the reward is bounded. For each experiment we set expectations and variances before truncation as $\mu_i' = 0.6, 0.5, 0.5, 0.4, 0.4$ and $\sigma_i = 0.4, 0.2, 0.4, 0.2, 0.4$. The expectation of each arm after truncation is given by $\mu_i = 0.519, 0.5, 0.5, 0.465, 0.481$ for support $[0, 1]$ and $\mu_i = 0.319, 0.390, 0.265, 0.320, 0.206$ for support $(-\infty, 1]$. We see from Fig. 3 that the policies for the nonparametric model work much better than that for the Bernoulli model.

6. Properties of $D_{\inf}$ in the Semi-bounded Support Model

In this section we extend some results on $D_{\inf}(F, \mu; \mathcal{A}_0)$ in Honda and Takemura (2010) to model $\mathcal{A} = \mathcal{A}_{-\infty}$ and prove Theorem 2.

The minimization function $D_{\inf}(F, \mu; \mathcal{A})$ is expressed as

$$\begin{aligned} \text{minimize: } & \int \left(\log \frac{\mathrm{d}F}{\mathrm{d}G} \right) \mathrm{d}F \\ \text{subject to: } & G \in \mathcal{A} \text{ is a positive finite measure on } (-\infty, 1], \\ & \int \mathrm{d}G = 1, \ \int x \mathrm{d}G > \mu \,, \end{aligned}$$

which has an infinite-dimensional variable and finite constraints. An optimization problem of this form is called a *partially-finite convex optimization* and many researches have been conducted (Borwein and Lewis, 1993; Ito et al., 2000). We can prove the relation $D_{\inf}(F, \mu; \mathcal{A}_0) = L_{\max}(F, \mu)$ in Prop. 1 in a generic way for this problem although it is proved in a problem-specific way in Honda and Takemura (2010, Theorem 5). Nevertheless, we were not able to find a result straightforwardly applicable to our target $D_{\inf}(F, \mu; \mathcal{A})$ for the reason below and we analyze this problem in a problem-specific way.

The difficulty in the model $\mathcal{A}$ lies in the fact that $\mathcal{A}$ is not compact and the operator $x : \mathcal{A} \to \mathbb{R} : G \mapsto \int x \mathrm{d}G$ in the constraint is not continuous under the Lévy metric since $f(x) = x$ is not a bounded function on $(-\infty, a]$. For this reason it is necessary to evaluate the effect of tail weights of measures on expectations precisely.

First we consider the function $L(\nu; F, \mu) = \mathrm{E}_F[\log(1 - (X - \mu))\nu]$. The integrand $l(x, \nu) \equiv \log(1 - (x - \mu)\nu)$ is differentiable in $\nu \in (0, (1 - \mu)^{-1})$ for all $x \in (-\infty, 1]$ with

$$\frac{\partial l(x, \nu)}{\partial \nu} = -\frac{x - \mu}{1 - (x - \mu)\nu} = \frac{1}{\nu}\left(1 - \frac{1}{1 - (x - \mu)\nu}\right),$$
$$\frac{\partial^2 l(x, \nu)}{\partial \nu^2} = -\frac{(x - \mu)^2}{(1 - (x - \mu)\nu)^2}.$$

Since they are bounded in $x \in (-\infty, 1]$, the integral $L(\nu; F, \mu)$ is differentiable in ν with

$$L'(\nu; F, \mu) \equiv \frac{\partial L(\nu; F, \mu)}{\partial \nu} = \frac{1}{\nu}\left(1 - \mathrm{E}_F\left[\frac{1}{1 - (X - \mu)\nu}\right]\right), \tag{12}$$
$$L''(\nu; F, \mu) \equiv \frac{\partial^2 L(\nu; F, \mu)}{\partial \nu^2} = -\mathrm{E}_F\left[\frac{(X - \mu)^2}{(1 - (X - \mu)\nu)^2}\right]. \tag{13}$$

From these derivatives the optimal solution $\nu^* = \nu^*(F, \mu) = \mathrm{argmax}_{0 \le \nu \le (1-\mu)^{-1}} L(\nu; F, \mu)$ of (2) exists uniquely except for the case $X = \mu$ (a.s.) and satisfies the properties in the following lemmas.

Lemma 6 *Assume that* $\mathrm{E}(F) < \mu < 1$ *holds. If* $\mathrm{E}_F[(1 - \mu)/(1 - X)] < 1$ *then* $\nu^* = (1 - \mu)^{-1}$ *and therefore* $\mathrm{E}_F[1/(1 - (X - \mu)\nu^*)] < 1$. *Otherwise,* $\nu^* \in (0, (1 - \mu)^{-1})$ *and* $\mathrm{E}_F[1/(1 - (X - \mu)\nu^*)] = 1$.

Lemma 7 $L_{\max}(F, \mu)$ *is differentiable in* $\mu < \mathrm{E}(F)$ *with*

$$\frac{\mathrm{d}L_{\max}(F, \mu)}{\mathrm{d}\mu} = \nu^*(F, \mu) \le \frac{1}{1 - \mu}.$$

Lemma 6 is straightforward from the derivatives (12) and (13). The proof of Lemma 7 is completely analogous to the proof of Honda and Takemura (2011, Theorems 3 (iii)) where the same results is derived for distributions on a finite support. We give the proof for completeness in Appendix B.

Define $F_{(a)} \in \mathcal{A}_a$ as the distribution obtained by transferring the probability of $(-\infty, a)$ under F to $x = a$, that is, the cumulative distribution function of $F_{(a)}$ is defined as

$$\bar{F}_{(a)}(x) \equiv \begin{cases} 0 & x < a, \\ \bar{F}(x) & x \ge a. \end{cases}$$

Recall that $L_{\max}(F, \mu) = \max_{0 \le \nu \le (1-\mu)^{-1}} L(\nu; F, \mu) = \max_{0 \le \nu \le (1-\mu)^{-1}} \mathrm{E}_F[\log(1 - (X - \mu))\nu]$. Now we give the key to extension for the semi-bounded support in the following lemma, which shows that the effect of the tail weight is bounded uniformly if the expectation is bounded from below.

Lemma 8 *Fix arbitrary $\mu, \tilde{\mu} < 1$ and $\epsilon > 0$. Then there exists $a(\epsilon, \mu, \tilde{\mu})$ such that $|L_{\max}(F_{(a)}, \mu) - L_{\max}(F, \mu)| \leq \epsilon$ for all $a \leq a(\epsilon, \mu, \tilde{\mu})$ and all $F \in \mathcal{A}$ such that $\mathrm{E}(F) \geq \tilde{\mu}$.*

Proof Take sufficiently small $a < \min\{0, \mu\}$ and define $A = (-\infty, a), B = [a, 1]$. Note that $F(A) + F(B) = 1$. First we have

$$F(A) \leq \frac{1-\tilde{\mu}}{1-a} \tag{14}$$

$$\int_A x \mathrm{d}F(x) \geq \tilde{\mu} - 1 + F(A) \tag{15}$$

from

$$\mathrm{E}(F) \leq aF(A) + 1 \cdot F(B) = 1 - (1-a)F(A)$$
$$\mathrm{E}(F) \leq \int_A x \mathrm{d}F(x) + 1 \cdot F(B)\,,$$

respectively. Next, $L_{\max}(F, \mu)$ can be written as

$$\begin{aligned} L_{\max}(F, \mu) &= \max_{0 \leq \nu \leq \frac{1}{1-\mu}} \mathrm{E}_F[\log(1 - (X-\mu)\nu)] \\ &= \max_{0 \leq \nu \leq \frac{1}{1-\mu}} \left\{ \int_A \log \frac{1-(x-\mu)\nu}{1-(a-\mu)\nu} \mathrm{d}F(x) + \int_B \log(1-(x-\mu)\nu) \mathrm{d}F_{(a)}(x) \right\}. \end{aligned} \tag{16}$$

Since $(1-(x-\mu)\nu)/(1-(a-\mu)\nu)$ is increasing in ν for $x \leq a$, substituting 0 and $(1-\mu)^{-1}$ into ν, we can bound the first term as

$$\begin{aligned} 0 &\leq \int_A \log \frac{1-(x-\mu)\nu}{1-(a-\mu)\nu} \mathrm{d}F(x) \\ &\leq \int_A \log \frac{1-x}{1-a} \mathrm{d}F(x) \\ &\leq F(A) \int_A \log(1-x) \frac{\mathrm{d}F(x)}{F(A)} && \text{(by } a \leq 0\text{)} \\ &\leq F(A) \log \left(\int_A (1-x) \frac{\mathrm{d}F(x)}{F(A)} \right) && \text{(Jensen's inequality)} \\ &\leq F(A) \log \frac{1-\tilde{\mu}}{F(A)}\,. && \text{(by (15))} \end{aligned}$$

From $\lim_{x \to 0} x \log x = 0$ and (14), the first term of (16) converges to 0 as $a \to -\infty$. The second term of (16) equals $L_{\max}(F_{(a)}, \mu)$ and the proof is completed. ■

Now we show Theorem 2 based on the preceding lemmas.

Proof of Theorem 2 (i) Recall that $G_{(a)}$ is the distribution such that the weight of G on $(-\infty, a)$ is transported to the point a. Thus, if $F \in \mathcal{A}_a$ is absolutely continuous with respect to G then $\mathrm{d}F/\mathrm{d}G \geq \mathrm{d}F/\mathrm{d}G_{(a)}$ holds almost everywhere on the support of F and we have $D(F\|G) \geq D(F\|G_{(a)})$. On the other hand if F is not absolutely continuous then

$D(F\|G) = \infty$ and therefore $D(F\|G) \geq D(F\|G_{(a)})$ still holds for this case. Combining them we have

$$\begin{aligned}\inf_{G\in\mathcal{A}:\mathrm{E}(G)>\mu} D(F\|G) &\geq \inf_{G\in\mathcal{A}:\mathrm{E}(G)>\mu} D(F\|G_{(a)}) \\ &\geq \inf_{G\in\mathcal{A}:\mathrm{E}(G_{(a)})>\mu} D(F\|G_{(a)}) \quad \big(\text{by } \mathrm{E}(G) \leq \mathrm{E}(G_{(a)})\big) \\ &= \inf_{G\in\mathcal{A}_a:\mathrm{E}(G)>\mu} D(F\|G)\,.\end{aligned}$$

On the other hand it holds from $\mathcal{A}_a \subset \mathcal{A}$ that

$$\inf_{G\in\mathcal{A}:\mathrm{E}(G)>\mu} D(F\|G) \leq \inf_{G\in\mathcal{A}_a:\mathrm{E}(G)>\mu} D(F\|G)$$

and we obtain $\inf_{G\in\mathcal{A}:\mathrm{E}(G)>\mu} D(F\|G) = \inf_{G\in\mathcal{A}_a:\mathrm{E}(G)>\mu} D(F\|G)$.

(ii) We show $D_{\inf}(F,\mu;\mathcal{A}) \leq L_{\max}(F,\mu)$ and $D_{\inf}(F,\mu;\mathcal{A}) \geq L_{\max}(F,\mu)$ separately. To prove the former inequality, let us consider a measure for any (measurable) set $S \subset \mathbb{R}$

$$G^*(S) \equiv \begin{cases} \int_S \frac{1-\mu}{1-x}\mathrm{d}F + (1 - \mathrm{E}_F[\frac{1-\mu}{1-X}])\mathbb{1}[1 \in S]\,, & \mathrm{E}_F[\frac{1-\mu}{1-X}] \leq 1\,, \\ \int_S \frac{1}{1-(x-\mu)\nu^*}\mathrm{d}F\,, & \mathrm{E}_F[\frac{1-\mu}{1-X}] > 1\,. \end{cases}$$

We can see from Lemma 6 that G^* is a probability measure such that $\mathrm{E}(G^*) = \mu$ and $D(F\|G^*) = L(\nu^*;F,\mu) = L_{\max}(F,\mu)$. Therefore the mixture distribution $(1-\epsilon)G^* + \epsilon\delta_1$ satisfies $\mathrm{E}((1-\epsilon)G^* + \epsilon\delta_1) = (1-\epsilon)\mu + \epsilon > \mu$ for any $\epsilon \in (0,1)$ where δ_1 is the point mass measure at 1. As a result,

$$\begin{aligned} D_{\inf}(F,\mu;\mathcal{A}) &\leq D(F\|(1-\epsilon)G^* + \epsilon\delta_1) \\ &\leq \int \log \frac{\mathrm{d}F}{\mathrm{d}((1-\epsilon)G^*)}\mathrm{d}F \\ &= D(F\|G^*) - \log(1-\epsilon) \\ &= L_{\max}(F,\mu) - \log(1-\epsilon) \end{aligned}$$

and we obtain $D_{\inf}(F,\mu;\mathcal{A}) \leq L_{\max}(F,\mu)$ by letting $\epsilon \downarrow 0$.

Next we show the latter inequality. Let $A = (-\infty, a]$ and $B = (a, 1]$, and define F_A and G_A as probability measures such that $F_A(S) = F(S \cap A)/F(A)$ and $G_A(S) = G(S \cap A)/G(A)$. Then, for any probability measure G such that F is absolutely continuous with respect to G, it holds that

$$\begin{aligned} D(F\|G) &= \int_A \log \frac{\mathrm{d}F}{\mathrm{d}G}\mathrm{d}F + \int_B \log \frac{\mathrm{d}F}{\mathrm{d}G}\mathrm{d}F \\ &= F(A)\int_A \log \frac{G(A)}{F(A)}\frac{\mathrm{d}F_A}{\mathrm{d}G_A}\mathrm{d}F_A + \int_B \log \frac{\mathrm{d}F}{\mathrm{d}G}\mathrm{d}F \\ &= F(A)\int_A \log \frac{G(A)}{F(A)}\mathrm{d}F_A + F(A)\int_A \log \frac{\mathrm{d}F_A}{\mathrm{d}G_A}\mathrm{d}F_A + \int_B \log \frac{\mathrm{d}F}{\mathrm{d}G}\mathrm{d}F \\ &= F(A)\log \frac{G(A)}{F(A)} + F(A)D(F_A\|G_A) + \int_B \log \frac{\mathrm{d}F}{\mathrm{d}G}\mathrm{d}F \\ &\geq F(A)\log \frac{G(A)}{F(A)} + \int_B \log \frac{\mathrm{d}F}{\mathrm{d}G}\mathrm{d}F \\ &= D(F_{(a)}\|G_{(a)}) \end{aligned}$$

and therefore,

$$\begin{aligned}\inf_{G\in\mathcal{A}:\mathrm{E}(G)>\mu} D(F\|G) &\geq \inf_{G\in\mathcal{A}:\mathrm{E}(G)>\mu} D(F_{(a)}\|G_{(a)}) \\ &\geq \inf_{G\in\mathcal{A}_a:\mathrm{E}(G_{(a)})>\mu} D(F_{(a)}\|G_{(a)}) \quad (\text{by } \mathrm{E}(G)\leq \mathrm{E}(G_{(a)}))\,.\end{aligned}$$

Let $F'_{(a)}$ and $G'_{(a)}$ be the probability distributions of $(X-a)/(1-a)$ when X follows $F_{(a)}$ and $G_{(a)}$, respectively. Then, letting $\epsilon>0$ be arbitrary and $a<\mu$ be sufficiently small, we obtain from invariance of KL divergence under scale transformation that

$$\begin{aligned}\inf_{G\in\mathcal{A}:\mathrm{E}(G)>\mu} D(F\|G) &\geq \inf_{G\in\mathcal{A}:\mathrm{E}(G_{(a)})>\mu} D(F_{(a)}\|G_{(a)}) \\ &= \inf_{G\in\mathcal{A}:\mathrm{E}(G'_{(a)})>\frac{\mu-a}{1-a}} D(F'_{(a)}\|G'_{(a)}) \\ &= D_{\inf}\left(F'_{(a)}, \frac{\mu-a}{1-a}; \mathcal{A}_0\right) \\ &= L_{\max}\left(F'_{(a)}, \frac{\mu-a}{1-a}\right) \qquad (\text{by Prop. 1}) \\ &= L_{\max}\left(F_{(a)}, \mu\right) \quad (\text{by expression of } L_{\max} \text{ in (2)}) \\ &\geq L_{\max}(F,\mu)-\epsilon \qquad (\text{by Lemma 8})\end{aligned}$$

and we complete the proof by letting $\epsilon\downarrow 0$. ■

7. Large Deviation Probabilities for Empirical Distributions Measured with $D_{\inf}$

It is essential for evaluation of IMED to derive large deviation probabilities on $\hat{F}_{i,t}$ and $\hat{\mu}_{i,t}$. In this section we discuss probabilities on the empirical distribution and the mean from a generic distribution $F\in\mathcal{A}$, for which we write $(\hat{F}_t,\hat{\mu}_t)$ by dropping the subscript i from $(\hat{F}_{i,t},\hat{\mu}_{i,t})$.

The key to the non-asymptotic evaluation lies in the fact that

$$\begin{aligned}D_{\inf}(\hat{F}_t,\mu) &= \max_{0\leq\nu\leq\frac{1}{1-\mu}} \mathrm{E}_{\hat{F}_t}[\log(1-(X-\mu)\nu)] \\ &= \max_{0\leq\nu\leq\frac{1}{1-\mu}} \left\{\frac{1}{t}\sum_{l=1}^{t}\log(1-(X_l-\mu)\nu)\right\},\end{aligned}$$

where each X_l follows distribution F. Although it involves a maximization, it is essentially an empirical mean of one-dimensional random variables $\log(1-(X_l-\mu)\nu)$. By Cramér's theorem below, we can bound the large deviation probability for such an empirical mean in a non-asymptotic form.

Proposition 9 (Dembo and Zeitouni, 1998, Eqs. (2.2.12) and (2.2.13)) *Assume that the moment generating function* $\mathrm{E}_F[\mathrm{e}^{\lambda X}]$ *exists in some neighborhood of* $\lambda=0$. *Then, for*

any $x \in \mathbb{R}$

$$\frac{1}{t}\log P_F[\hat{\mu}_t \geq x] \leq -\sup_{\lambda \geq 0}\left\{\lambda x - \log \mathrm{E}_F[\mathrm{e}^{\lambda X}]\right\}.$$

Also, if $x < \mathrm{E}(F)$ *then*

$$\frac{1}{t}\log P_F[\hat{\mu}_t \leq x] \leq -\Lambda^*(x) \tag{17}$$

and if $x > \mathrm{E}(F)$ *then*

$$\frac{1}{t}\log P_F[\hat{\mu}_t \geq x] \leq -\Lambda^*(x) \tag{18}$$

where $\Lambda^*(x) = \sup_\lambda\{\lambda x - \log \mathrm{E}_F[\mathrm{e}^{\lambda X}]\}$.

We prove Props. 10–12 given below by Cramér's theorem.

Proposition 10 *For any* $F \in \mathcal{A}, \mu > \mathrm{E}(F)$ *and* $u < D_{\inf}(F, \mu)$,

$$P_F[D_{\inf}(\hat{F}_t, \mu) \leq u] \leq \mathrm{e}^{-t\tilde{\Lambda}^*(u)},$$

where $\tilde{\Lambda}^*(x) = \sup_\lambda\{\lambda x - \mathrm{E}_F[(1-(X-\mu)\nu^*)^\lambda]\}$ *for* $\nu^* = \mathrm{argmax}_{0\leq\nu\leq(1-\mu)^{-1}} \mathrm{E}_F[\log(1-(X-\mu)\nu)]$.

Proof For $\nu^* = \mathrm{argmax}_{0\leq\nu\leq(1-\mu)^{-1}} \mathrm{E}_F[\log(1-(X-\mu)\nu)]$ we have

$$\begin{aligned} P_F[D_{\inf}(\hat{F}_t, \mu) \leq u] &= P_F\left[\max_{0\leq\nu\leq(1-\mu)^{-1}} \mathrm{E}_{\hat{F}_t}[\log(1-(X-\mu)\nu)] \leq u\right] \\ &\leq P_F\left[\mathrm{E}_{\hat{F}_t}[\log(1-(X-\mu)\nu^*)] \leq u\right]. \end{aligned}$$

For $X_1, X_2, \cdots$ following distribution F, we can regard $\mathrm{E}_{\hat{F}_t}[\log(1-(X-\mu)\nu^*)]$ as the empirical mean of $Y_i = \log(1-(X_i-\mu)\nu^*)$, $i = 1, \cdots, t$, which has expectation $D_{\inf}(F, \mu)$. Then the theorem follows immediately from (17) of Prop. 9. ■

Proposition 11 *Fix any* $F \in \mathcal{A}$ *and* $\mu < \mathrm{E}(F)$ *and assume that the moment generating function* $\mathrm{E}_F[\mathrm{e}^{\lambda X}]$ *of* F *exists in some neighborhood of* $\lambda = 0$. *(i) For* $\lambda_\mu = \sup\{\lambda \in \mathbb{R}\cup\{+\infty\} : \mathrm{E}_F[((1-X)/(1-\mu))^\lambda] \leq 1\}$, *we have* $\lambda_\mu > 1$. *(ii) For any* $u \in \mathbb{R}$,

$$P_F[D_{\inf}(\hat{F}_t, \mu) \geq u,\ \hat{\mu}_t \leq \mu] \ \leq\ \begin{cases} \mathrm{e}^{-t\Lambda^*(\mu)}, & \text{if } u \leq \Lambda^*(\mu)/\lambda_\mu, \\ 2\mathrm{e}(1+\lambda_\mu t)\mathrm{e}^{-t\lambda_\mu u}, & \text{otherwise.} \end{cases}$$

where $\Lambda^*(x) = \sup_\lambda\{\lambda x - \log \mathrm{E}_F[\mathrm{e}^{\lambda X}]\}$ *and we define* $\lambda\mathrm{e}^{-\lambda} = 0$ *for* $\lambda = +\infty$.

Remark 1 Since $D_{\inf}(\hat{F}_t, \mu) \geq u$ implies

$$\begin{aligned} D(\hat{F}_t \| F) &\geq D_{\inf}(\hat{F}_t, \mathrm{E}(F)) \\ &\geq D_{\inf}(\hat{F}_t, \mu) \\ &\geq u\,, \end{aligned}$$

it is easy to prove from Sanov's theorem (Dembo and Zeitouni, 1998, Chap. 6.2) that

$$\limsup_{t\to\infty} \frac{1}{t} \log P_F[D_{\inf}(\hat{F}_t, \mu) \geq u,\ \hat{\mu}_t \leq \mu] \leq -u\,,$$

that is, $P_F[D_{\inf}(\hat{F}_t, \mu) \geq u,\ \hat{\mu}_t \leq \mu]$ is roughly bounded by e^{-tu}. Prop. 11 shows that this bound can be refined to $\mathrm{e}^{-t\lambda_\mu u}$ for large u and its coefficient is explicitly bounded by a polynomial $2\mathrm{e}(1+\lambda_\mu t)$.

Proof of Proposition 11 (i) Since we assume $\mathrm{E}[\mathrm{e}^{\lambda X}] < \infty$ in some neighborhood of $\lambda = 0$,

$$\mathrm{E}_F\left[\left(\frac{1-X}{1-\mu}\right)^{\lambda}\right] = \frac{\mathrm{E}_F[(1-X)^\lambda]}{(1-\mu)^\lambda}$$

is finite and continuous in $\lambda \geq 0$. We obtain $\lambda_\mu > 1$ from

$$\mathrm{E}_F\left[\left(\frac{1-X}{1-\mu}\right)^{1}\right] = \frac{1-\mathrm{E}(F)}{1-\mu} < 1\,.$$

(ii) Fix an arbitrary $\delta > 0$ and let $M_\delta = \lceil 1/(2\delta(1-\mu))\rceil$. Define $\nu_{(m)}$ for $m = -M_\delta, -M_\delta+1, \cdots, 0, \cdots, M_\delta$ by

$$\nu_{(m)} = \frac{1 + \frac{m}{M_\delta}}{2(1-\mu)}\,.$$

Then $\{[\nu_{(m)}, \nu_{(m+1)}]\}_{m=-M_\delta,\cdots,M_\delta-1}$ partitions $[0, (1-\mu)^{-1}]$ into intervals with length at most δ. Therefore the event $\{D_{\inf}(\hat{F}_t, \mu) \geq u\}$ can be expressed as

$$\begin{aligned} \{D_{\inf}(\hat{F}_t, \mu) \geq u\} &= \left\{\exists \nu \in \left[0, \tfrac{1}{1-\mu}\right],\ L(\nu; \hat{F}_t, \mu) \geq u\right\} \\ &= \bigcup_{m=-M_\delta}^{-1} \left\{\exists \nu \in \left[\nu_{(m)}, \nu_{(m+1)}\right],\, L(\nu; \hat{F}_t, \mu) \geq u\right\} \\ &\quad \cup \bigcup_{m=1}^{M_\delta} \left\{\exists \nu \in \left[\nu_{(m-1)}, \nu_{(m)}\right],\, L(\nu; \hat{F}_t, \mu) \geq u\right\}. \end{aligned} \tag{19}$$

Since $|\nu_{(m+1)} - \nu_{(m)}| \leq \delta$ and $L(\nu; \hat{F}_t, \mu)$ is concave in ν, it holds for $m \leq -1$ that

$$\begin{aligned} &\left\{\exists \nu \in \left[\nu_{(m)}, \nu_{(m+1)}\right],\, L(\nu; \hat{F}_t, \mu) \geq u\right\} \\ &\quad \subset \left\{L(\nu_{(m+1)}; \hat{F}_t, \mu) - \delta \min\{0, L'(\nu_{(m+1)}; \hat{F}_t, \mu)\} \geq u\right\} \\ &\quad \subset \left\{L(\nu_{(m+1)}; \hat{F}_t, \mu) - \delta \min\{0, L'(\nu_{(0)}; \hat{F}_t, \mu)\} \geq u\right\}. \end{aligned} \tag{20}$$

Similarly it holds for $m \geq 1$ that

$$\begin{aligned}&\left\{\exists \nu \in \left[\nu_{(m-1)}, \nu_{(m)}\right],\, L(\nu; \hat{F}_t, \mu) \geq u\right\} \\ &\quad \subset \left\{L(\nu_{(m-1)}; \hat{F}_t, \mu) + \delta \max\{0, L'(\nu_{(0)}; \hat{F}_t, \mu)\} \geq u\right\}. \end{aligned} \tag{21}$$

Here the derivative L' is expressed from (12) as

$$L'(\nu; \hat{F}_t, \mu) = \frac{1}{\nu} - \frac{1}{\nu} \mathrm{E}_{\hat{F}_t}\left[\frac{1}{1-(X-\mu)\nu}\right].$$

Since $1/(1-(x-\mu)\nu)$ is positive and increasing in $x \leq 1$, it is bounded as

$$\frac{1}{\nu} \geq L'(\nu; \hat{F}_t, \mu) \geq \frac{1}{\nu} - \frac{1}{\nu}\frac{1}{1-(1-\mu)\nu} = -\frac{1-\mu}{1-(1-\mu)\nu}.$$

Thus $L'(\nu_{(0)}; \hat{F}_t, \mu) = L'(1/(2(1-\mu)); \hat{F}_t, \mu)$ is bounded as

$$2(1-\mu) \geq L'(\nu_{(0)}; \hat{F}_t, \mu) \geq -2(1-\mu).$$

Combining this with (19), (20) and (21) we obtain

$$P_F[D_{\inf}(\hat{F}_t, \mu) \geq u] \leq \sum_{\substack{m \neq 0: \\ -M_\delta \leq m \leq M_\delta}} P_F\Big[L(\nu_{(m)}; \hat{F}_t, \mu) \geq u - 2(1-\mu)\delta\Big]. \tag{22}$$

Now recall that

$$\lambda_\mu = \sup\left\{\lambda : \mathrm{E}_F\left[\left(\frac{1-X}{1-\mu}\right)^\lambda\right] \leq 1\right\} > 1.$$

Then, by Prop. 9,

$$\begin{aligned}&P_F\Big[L(\nu_{(m)}; \hat{F}_t, \mu) \geq u - 2(1-\mu)\delta\Big] \\ &\leq \exp\left(-t \sup_{\lambda \geq 0}\left\{\lambda(u - 2(1-\mu)\delta) - \log \mathrm{E}_F[\mathrm{e}^{\lambda \log(1-(X-\mu)\nu_{(m)})}]\right\}\right) \\ &\leq \exp\Bigg(-t \sup_{\lambda \geq 1}\Big\{\lambda(u - 2(1-\mu)\delta) \\ &\qquad\qquad - \log\left(\mathrm{E}_F[\mathrm{e}^{\lambda \log(1-(X-\mu)\cdot 0)}] \vee \mathrm{E}_F[\mathrm{e}^{\lambda \log(1-(X-\mu)\cdot(1-\mu)^{-1})}]\right)\Big\}\Bigg) \qquad (23) \\ &= \exp\left(-t \sup_{\lambda \geq 1}\left\{\lambda(u - 2(1-\mu)\delta) - \log\left(1 \vee \mathrm{E}_F\left[\left(\frac{1-X}{1-\mu}\right)^\lambda\right]\right)\right\}\right) \\ &\leq \exp\left(-t\lambda_\mu(u - 2(1-\mu)\delta)\right), \qquad (24)\end{aligned}$$

where (23) follows from $0 \leq \nu_{(m)} \leq (1-\mu)^{-1}$ and the convexity of $\mathrm{E}_F[\mathrm{e}^{\lambda \log(1-(X-\mu)\nu)}]$ in $\nu \in [0, (1-\mu)^{-1}]$ for $\lambda \geq 1$. Therefore we obtain from (22) and (24) that

$$\begin{aligned}P_F[D_{\inf}(\hat{F}_t, \mu) \geq u] &\leq 2M_\delta \exp\left(-t\lambda_\mu(u - 2(1-\mu)\delta)\right) \\ &\leq 2\left(1 + \frac{1}{2(1-\mu)\delta}\right)\exp\left(-t\lambda_\mu(u - 2(1-\mu)\delta)\right)\end{aligned}$$

and we complete the proof by letting $\delta = 1/(2t\lambda_\mu(1-\mu))$ and combining it with (17). ■

We prove Theorem 3 by the above two propositions. We also use the following proposition on the large deviation probability of $D_{\inf}(\hat{F}_t, \mu)$ under a more general setting for the proof of Theorem 5.

Proposition 12 *Fix any $u, \mu \in \mathbb{R}$ and $F \in \mathcal{A}$ such that $\mathrm{E}(F) < \mu < 1$. Then*

$$P_F[D_{\inf}(\hat{F}_t, \mu) \geq u] \leq 2\mathrm{e}(1+t)\exp\left(-t\left(u - \log\frac{1-\mathrm{E}(F)}{1-\mu}\right)\right).$$

Proof Since (22) and (23) also hold for the case of this theorem, we obtain the theorem by letting $\lambda = 1$ and $\delta = 1/(2t(1-\mu))$. ■

8. Regret Analysis for IMED

In this section we prove Theorem 3 by using a technique similar to that for UCB policies. First we prove Lemma 13 below as a fundamental property of the IMED policy on the minimum index $I^*(l) \equiv \min_{i\in\{1,2,\cdots,K\}} I_i(l)$.

Lemma 13 *For any $x > 0$ and arm i,*

$$\sum_{l=1}^{\infty} \mathbb{1}[I^*(l) \leq x,\, J(l) = i] \leq \mathrm{e}^x.$$

Proof This is straightforward from

$$\begin{aligned}
\sum_{l=1}^{\infty} \mathbb{1}[I^*(l) \leq x,\, J(l) = i] &= \sum_{t=1}^{\infty}\sum_{l=1}^{\infty} \mathbb{1}[I^*(l) \leq x,\, J(l) = i,\, T_i(l) = t] \\
&\leq \sum_{t=1}^{\infty}\sum_{l=1}^{\infty} \mathbb{1}[\log t \leq x,\, J(l) = i,\, T_i(l) = t] \\
&\qquad\qquad (J(l) = i \text{ implies } I^*(l) = I_i(l) \geq \log T_i(l)) \\
&= \sum_{t=1}^{\lfloor \mathrm{e}^x \rfloor}\sum_{l=1}^{\infty} \mathbb{1}[J(l) = i,\, T_i(l) = t] \\
&\leq \sum_{t=1}^{\lfloor \mathrm{e}^x \rfloor} 1 \qquad\qquad (\{J(l) = i,\, T_i(l) = t\} \text{ occurs for at most one } l) \\
&\leq \mathrm{e}^x.
\end{aligned}$$

■

We prove Theorem 3 by Lemma 14 below.

Lemma 14 *It holds for any $\mu < \mu^*$ and arm i that*

$$\mathrm{E}\left[\sum_{l=1}^{\infty} \mathbb{1}[\hat{\mu}^*(l) \leq \mu,\, J(l) = i]\right] \leq \inf_{j\in\mathcal{I}_{\mathrm{opt}}}\left\{\frac{6\mathrm{e}}{(1-1/\lambda_{j,\mu})(1-\mathrm{e}^{-(1-1/\lambda_{j,\mu})\Lambda_j^*(\mu)})^3}\right\}.$$

Proof Let j be any optimal arm, that is, j such that $\Delta_j = 0$. We will bound the RHS of

$$\begin{aligned}\sum_{l=1}^{\infty} \mathbb{1}[\hat{\mu}^*(l) \le \mu,\, J(l) = i] &= \sum_{l=1}^{\infty} \mathbb{1}[\hat{\mu}_j(l) \le \hat{\mu}^*(l) \le \mu,\, J(l) = i] \\ &\le \sum_{t=1}^{\infty}\sum_{l=1}^{\infty} \mathbb{1}[\hat{\mu}_{j,t} \le \hat{\mu}^*(l) \le \mu,\, T_j(l) = t,\, J(l) = i]\,. \quad (25)\end{aligned}$$

Since $\{\hat{\mu}_{j,t} \le \hat{\mu}^*(l) \le \mu,\, T_j(l) = t\}$ implies

$$\begin{aligned} I^*(l) &= \min_i I_i(l) \\ &\le I_j(l) \\ &= tD_{\inf}(\hat{F}_{j,t}, \hat{\mu}^*(l)) + \log t \\ &\le tD_{\inf}(\hat{F}_{j,t}, \mu) + \log t\,, \end{aligned}$$

we see from Lemma 13 that $\{\hat{\mu}_{j,t} \le \hat{\mu}^*(l) \le \mu,\, T_j(l) = t,\, J(l) = i\}$ occurs for at most $t\mathrm{e}^{tD_{\inf}(\hat{F}_{j,t},\mu)}$ rounds. Therefore from (25) we obtain

$$\sum_{l=1}^{\infty} \mathbb{1}[\hat{\mu}^*(l) \le \mu,\, J(l) = i] \le \sum_{t=1}^{\infty} \mathbb{1}[\hat{\mu}_{j,t} \le \mu]\, t\mathrm{e}^{tD_{\inf}(\hat{F}_{j,t},\mu)}\,. \quad (26)$$

Let $P(u) \equiv P_{F_j}[D_{\inf}(\hat{F}_{j,t}, \mu) > u,\, \hat{\mu}_{j,t} \le \mu]$. Simply writing λ_j and Λ_j^* for $\lambda_{j,\mu}$ and $\Lambda_j^*(\mu)$ in (4) and (5), respectively, we have from Prop. 11 that

$$\begin{aligned} &\mathrm{E}\left[\mathbb{1}[\hat{\mu}_{j,t} \le \mu]\, t\mathrm{e}^{tD_{\inf}(\hat{F}_{j,t},\mu)}\right] \\ &= \int_0^{\infty} t\mathrm{e}^{tu}(-\mathrm{d}P(u)) \\ &= \left[t\mathrm{e}^{tu}(-P(u))\right]_0^{\infty} + \int_0^{\infty} t^2\mathrm{e}^{tu}P(u)\mathrm{d}u \qquad \text{(integration by parts)} \\ &\le t\mathrm{e}^{-t\Lambda_j^*} + \int_0^{\Lambda_j^*/\lambda_j} t^2\mathrm{e}^{tu}\cdot\mathrm{e}^{-t\Lambda_j^*}\mathrm{d}u + \int_{\Lambda_j^*/\lambda_j}^{\infty} t^2\mathrm{e}^{tu}\cdot 2\mathrm{e}(1+\lambda_j t)\mathrm{e}^{-t\lambda_j u}\mathrm{d}u \\ &= t\mathrm{e}^{-t\Lambda_j^*} + t\left[\mathrm{e}^{t(u-\Lambda_j^*)}\right]_0^{\Lambda_j^*/\lambda_j} - 2\mathrm{e}t(1+\lambda_j t)\left[\frac{\mathrm{e}^{-t(\lambda_j-1)u}}{\lambda_j - 1}\right]_{\Lambda_j^*/\lambda_j}^{\infty} \\ &= t\mathrm{e}^{-t(1-1/\lambda_j)\Lambda_j^*} + 2\mathrm{e}t(1+\lambda_j t)\frac{\mathrm{e}^{-t(1-1/\lambda_j)\Lambda_j^*}}{\lambda_j - 1} \\ &= \left(\frac{1 - 1/\lambda_j + 2\mathrm{e}/\lambda_j}{1 - 1/\lambda_j}\right)\cdot t\mathrm{e}^{-t(1-1/\lambda_j)\Lambda_j^*} + \frac{2\mathrm{e}}{1-1/\lambda_j}\cdot t^2\mathrm{e}^{-t(1-1/\lambda_j)\Lambda_j^*}\,. \quad (27)\end{aligned}$$

From (26), (27) and formulas

$$\begin{aligned}\sum_{t=1}^{\infty} t\mathrm{e}^{-rt} &\le \frac{1}{(1-\mathrm{e}^{-r})^2} \le \frac{1}{(1-\mathrm{e}^{-r})^3} \\ \sum_{t=1}^{\infty} t^2\mathrm{e}^{-rt} &\le \frac{2}{(1-\mathrm{e}^{-r})^3}\,,\end{aligned}$$

it holds that

$$\begin{aligned}
\mathrm{E}\left[\sum_{l=1}^{\infty}\mathbb{1}[\hat{\mu}^*(l)\le\mu,\,J(l)=i]\right] &\le \left(\frac{1+(2\mathrm{e}-1)/\lambda_j+4\mathrm{e}}{1-1/\lambda_j}\right)\frac{1}{(1-\mathrm{e}^{-t(1-1/\lambda_j)\Lambda_j^*})^3}\\
&\le \left(\frac{1+(2\mathrm{e}-1)+4\mathrm{e}}{1-1/\lambda_j}\right)\frac{1}{(1-\mathrm{e}^{-t(1-1/\lambda_j)\Lambda_j^*})^3}\\
&= \frac{6\mathrm{e}}{(1-1/\lambda_j)(1-\mathrm{e}^{-t(1-1/\lambda_j)\Lambda_j^*})^3}\,. \qquad (28)
\end{aligned}$$

We complete the proof by taking j which minimizes (28) over the optimal arms $j\in\mathcal{I}_{\mathrm{opt}}$. ■

Proof of Theorem 3 First we decompose $T_i(n)$ as

$$\begin{aligned}
T_i(n) &= \sum_{l=1}^{n}\mathbb{1}[J(l)=i]\\
&= \sum_{l=1}^{n}\mathbb{1}[J(l)=i,\,\hat{\mu}^*(l)\le\mu^*-\delta]+\sum_{l=1}^{n}\mathbb{1}[J(l)=i,\,\hat{\mu}^*(l)\ge\mu^*-\delta]\,. \qquad (29)
\end{aligned}$$

The summation of the second term of (29) is bounded as

$$\begin{aligned}
\sum_{l=1}^{n}\mathbb{1}[J(l)=i,\,\hat{\mu}^*(l)\ge\mu^*-\delta] &= \sum_{t=1}^{n}\mathbb{1}\left[\bigcup_{l=1}^{n}\{J(l)=i,\,T_i(l)=t,\,\hat{\mu}^*(l)\ge\mu^*-\delta\}\right]\\
&\le \sum_{t=1}^{n}\mathbb{1}\left[\bigcup_{l=1}^{n}\{I_i(l)=I^*(l),\,T_i(l)=t,\,\hat{\mu}^*(l)\ge\mu^*-\delta\}\right].
\end{aligned}$$

Note that $I^*(l)\le\max_{i:\hat{\mu}_i(l)=\hat{\mu}^*(l)}I_i(l)=\max_{i:\hat{\mu}_i(l)=\hat{\mu}^*(l)}\log T_i(l)\le\log n$ for all $l\le n$. Then we have

$$\begin{aligned}
&\mathrm{E}\left[\sum_{l=1}^{n}\mathbb{1}[J(l)=i,\,\hat{\mu}^*(l)\ge\mu^*-\delta]\right]\\
&\le \mathrm{E}\left[\sum_{t=1}^{n}\mathbb{1}\Big[tD_{\inf}(\hat{F}_{i,t},\mu^*-\delta)\le\log n\Big]\right] \qquad (\text{by } I^*(l)=I_i(l)\ge tD_{\inf}(\hat{F}_i(l),\hat{\mu}^*(l)))\\
&= \sum_{t=1}^{\infty}P_{F_i}\Big[tD_{\inf}(\hat{F}_{i,t},\mu^*-\delta)\le\log n\Big]\\
&= \sum_{t=1}^{\infty}P_{F_i}\left[t\left(D_{\inf}(\hat{F}_{i,t},\mu^*)-\int_{\mu^*-\delta}^{\mu^*}\frac{\mathrm{d}D_{\inf}(\hat{F}_{i,t},\mu)}{\mathrm{d}\mu}\bigg|_{\mu=u}\mathrm{d}u\right)\le\log n\right]\\
&\le \sum_{t=1}^{\infty}P_{F_i}\left[t\left(D_{\inf}(\hat{F}_{i,t},\mu^*)-\int_{\mu^*-\delta}^{\mu^*}\frac{\mathrm{d}u}{1-u}\right)\le\log n\right] \qquad (\text{by Lemma 7})\\
&\le \sum_{t=1}^{\infty}P_{F_i}\left[t\left(D_{\inf}(\hat{F}_{i,t},\mu^*)-\frac{\delta}{1-\mu^*}\right)\le\log n\right].
\end{aligned}$$

By letting

$$M = \left\lceil \frac{\log n}{D_{\inf}(F_i, \mu^*) - \frac{2\delta}{1-\mu^*}} \right\rceil ,$$

we have

$$\begin{aligned}
&\mathrm{E}\left[\sum_{l=1}^{n} \mathbb{1}[J(l) = i,\ \hat{\mu}^*(l) \geq \mu^* - \delta]\right] \\
&\leq M - 1 + \sum_{t=M}^{\infty} P_{F_i}\left[t\left(D_{\inf}(\hat{F}_{i,t}, \mu^*) - \frac{\delta}{1-\mu^*}\right) \leq \log n\right] \\
&\leq M - 1 + \sum_{t=M}^{\infty} P_{F_i}\left[M\left(D_{\inf}(\hat{F}_{i,t}, \mu^*) - \frac{\delta}{1-\mu^*}\right) \leq \log n\right] \\
&\leq M - 1 + \sum_{t=M}^{\infty} P_{F_i}\left[D_{\inf}(\hat{F}_{i,t}, \mu^*) \leq D_{\inf}(F_i, \mu^*) - \frac{\delta}{1-\mu^*}\right] \\
&\leq M - 1 + \sum_{t=M}^{\infty} \mathrm{e}^{-t\tilde{\Lambda}(D_{\inf}(F_i,\mu^*) - \frac{\delta}{1-\mu^*})} \qquad \text{(by Prop. 10)} \\
&\leq \frac{\log n}{D_{\inf}(F_i, \mu^*) - \frac{2\delta}{1-\mu^*}} + \frac{1}{1 - \mathrm{e}^{-\tilde{\Lambda}_i^*(D_{\inf}(F_i,\mu^*) - \frac{\delta}{1-\mu})}} .
\end{aligned}$$

On the other hand, we can bound the expectation of the first term of (29) by Lemma 14 with $\mu := \mu^* - \delta$, which completes the proof of the theorem. ■

9. Concluding Remarks and Discussion

We considered a nonparametric stochastic bandit where only the upper bound of the reward is known. We proved that the theoretical bound does not depend on the knowledge of the lower bound of the reward. We also showed that the bound can be achieved by the IMED policy, an indexed version of the DMED policy.

A future work is to examine whether the assumption on existence of moment generating functions $\mathrm{E}_{F_i}[\mathrm{e}^{\lambda X}]$ can be weakened to existence of moments $\mathrm{E}_{F_i}[X^m]$. In the analysis of IMED it is important to evaluate tail probabilities of $\hat{\mu}_{i,t}$ and $D_{\inf}(\hat{F}_{i,t}, \mu) = \max_{0 \leq \nu \leq (1-\mu)^{-1}} \mathrm{E}_{\hat{F}_{i,t}}[\log(1 - (X - \mu)\nu)]$. Although the latter one is more essential in the behavior of IMED, this only requires the existence of the moment $\mathrm{E}[\mathrm{e}^{\lambda \log(1-(X-\mu)\nu)}] = \mathrm{E}[(1 - (X - \mu)\nu)^{\lambda}]$ and we assumed the existence of $\mathrm{E}_{F_i}[\mathrm{e}^{\lambda X}]$ only for the evaluation of $\hat{\mu}_{i,t}$. Furthermore, in the most part of evaluations involving $\hat{\mu}_{i,t}$ it suffices to show that

$$\sum_{t=1}^{\infty} t^p \Pr[|\hat{\mu}_{i,t} - \mu_i| > \delta] < \infty \tag{30}$$

for some $p \geq 0$, which we can assure to hold only by assuming $\mathrm{E}_{F_i}[X^{2+p}] < \infty$ (Chow and Lai, 1975). From these reasons we conjecture that the assumption $\mathrm{E}[\mathrm{e}^{\lambda X}] < \infty$ can be weakened by using (30) but it remains as an open problem.

Acknowledgments

This work was supported by JSPS Grant-in-Aid for Scientific Research No. 26106506, 25220001.

Appendix A. Representations of Constants for Large Deviation Probabilities

In Theorem 3, $\lambda_{i,\mu}$, $\Lambda_i^*(x)$ and $\tilde{\Lambda}_i^*(x)$ in (4)–(6) are used in the constant term of the regret. We discuss explicit representations of them in this appendix.

First we evaluate $\Lambda_i^*(x)$ and $\tilde{\Lambda}_i^*(x)$, which are Legendre-Fenchel transforms of cumulant generating functions of random variables X and $Y = \log(1-(X-\mu^*)\nu_i^*)$, respectively, where X follows F_i. If the support of F_i is bounded from below by $a > -\infty$ then by Hoeffding's inequality (Hoeffding, 1963) we have

$$\Lambda_i^*(\mu_i + \delta) \geq \frac{2\delta^2}{(1+a)^2}\,.$$

Similarly, from $Y \in [\log(1-(1-\mu^*)\nu_i^*),\, \log(1-(a-\mu^*)\nu_i^*)]$

$$\tilde{\Lambda}_i^*(D_{\inf}(F_i,\mu^*) - \delta) \geq \frac{2\delta^2}{\left(\log \frac{1-(a-\mu^*)\nu_i^*}{1-(1-\mu^*)\nu_i^*}\right)^2}\,.$$

Furthermore, we can evaluate $\Lambda_i^*(\mu_i+\delta)$ and $\tilde{\Lambda}_i^*(D_{\inf}(F_i,\mu^*)-\delta)$ for general cases including $a = -\infty$ by the following lemma.

Lemma 15 *For sufficiently small $\delta > 0$,*

$$\Lambda_i^*(\mu_i + \delta) \geq \frac{\delta^2}{2\sigma_i^2} + \mathrm{o}(\delta^2)\,, \tag{31}$$

$$\tilde{\Lambda}_i^*(D_{\inf}(F_i,\mu^*) - \delta) \geq \frac{(1-\mu^*)\delta^2}{4(1-\mu_i)} + \mathrm{o}(\delta^2)\,, \tag{32}$$

where $\sigma_i^2 = \mathrm{E}_{F_i}[(X-\mu_i)^2]$ is the variance of F_i.

Proof Since the cumulant generating function of F_i is expressed as

$$\log \mathrm{E}_{F_i}[\mathrm{e}^{\lambda X}] = \mu_i\lambda + \frac{\sigma_i^2\lambda^2}{2} + \mathrm{o}(\lambda^2)\,,$$

we obtain (31) from

$$\begin{aligned}
\Lambda_i^*(\mu_i+\delta) &= \sup_\lambda \left\{(\mu_i+\delta)\lambda - \log \mathrm{E}_{F_i}[\mathrm{e}^{\lambda X}]\right\} \\
&= \sup_\lambda \left\{\delta\lambda - \frac{\sigma_i^2\lambda^2}{2} + \mathrm{o}(\lambda^2)\right\} \\
&\geq \frac{\delta^2}{2\sigma_i^2} + \mathrm{o}(\delta^2)\,. \qquad \left(\text{by letting } \lambda := \delta/\sigma_i^2\right)
\end{aligned}$$

Similarly, from $\mathrm{E}_{F_i}[Y] = D_{\inf}(F_i, \mu^*)$ we have

$$\begin{aligned}\tilde{\Lambda}_i^*(D_{\inf}(F_i,\mu^*) - \delta) &= \sup_{\lambda} \left\{ (D_{\inf}(F_i,\mu^*) - \delta)\lambda - \log \mathrm{E}_{F_i}[\mathrm{e}^{\lambda Y}] \right\} \\ &\geq \frac{\delta^2}{2\tilde{\sigma}_i^2} + \mathrm{o}(\delta^2)\,, \end{aligned} \tag{33}$$

where $\tilde{\sigma}_i^2$ is the variance of $Y = \log(1 - (X - \mu^*)\nu_i^*)$. Since Y has expectation $\mathrm{E}_{F_i}[Y] = D_{\inf}(F_i, \mu^*)$, the variance $\tilde{\sigma}_i^2$ is expressed as

$$\begin{aligned}\tilde{\sigma}_i^2 &= \mathrm{E}_{F_i}[(Y - D_{\inf}(F_i,\mu^*))^2] \\ &= \mathrm{E}_{F_i}\left[\left(\log \frac{\mathrm{e}^Y}{\mathrm{e}^{D_{\inf}(F_i,\mu^*)}} \right)^2 \right]. \end{aligned}$$

Note that $(\log z)^2$ is smaller than z^{-1} for $z \to +0$ and smaller than z for $z \to \infty$. Thus there exists $c_0 > 0$ such that $(\log z)^2 \leq c_0(z + z^{-1})$ for all $z > 0$. In fact, this inequality holds for $c_0 \geq 0.533$ (and thus, for $c_0 = 1$). Therefore

$$\begin{aligned}\tilde{\sigma}_i^2 &\leq \mathrm{E}_{F_i}\left[\frac{\mathrm{e}^Y}{\mathrm{e}^{D_{\inf}(F_i,\mu^*)}} + \frac{\mathrm{e}^{D_{\inf}(F_i,\mu^*)}}{\mathrm{e}^Y} \right] \\ &\leq \mathrm{E}_{F_i}[\mathrm{e}^Y] + \mathrm{e}^{D_{\inf}(F_i,\mu^*)}\mathrm{E}_{F_i}[\mathrm{e}^{-Y}] && \text{(by } D_{\inf}(F_i,\mu^*) \geq 0) \\ &= \mathrm{E}_{F_i}[\mathrm{e}^Y] + \mathrm{e}^{\mathrm{E}_{F_i}[Y]}\mathrm{E}_{F_i}[\mathrm{e}^{-Y}] \\ &\leq \mathrm{E}_{F_i}[\mathrm{e}^Y] + \mathrm{E}_{F_i}[\mathrm{e}^Y]\mathrm{E}_{F_i}[\mathrm{e}^{-Y}] && \text{(by Jensen's inequality)} \\ &= (1 - (\mu_i - \mu^*)\nu_i^*) \cdot \left(1 + \mathrm{E}_{F_i}\left[\frac{1}{1 - (X - \mu^*)\nu_i^*} \right] \right) \\ &\leq \left(1 - \frac{\mu_i - \mu^*}{1 - \mu^*} \right) \cdot (1 + 1) && \text{(by Lemma 6)} \\ &= \frac{2(1 - \mu_i)}{1 - \mu^*}\,. \end{aligned} \tag{34}$$

We obtain (32) by combining (34) with (33). ■

Next we bound $\lambda_{i,\mu}$ with an explicit form in the following lemma and we see that $\lambda_{i,\mu_i - \delta} \geq 1 + (1 - \mu_i)\delta/\sigma_i^2 + \mathrm{o}(\delta)$.

Lemma 16 *If $\mu < \mu_i < 1$ then*

$$\lambda_{i,\mu} \geq \begin{cases} 1 + \frac{(1-\mu)(\mu_i - \mu)}{\sigma_i^2 - (1-\mu_i)(\mu_i - \mu)}, & \text{if } \sigma_i^2 \geq (\mu_i - \mu)(2 - \mu_i - \mu), \\ 2, & \text{otherwise.} \end{cases} \tag{35}$$

Proof Since x^λ is convex in λ, we have

$$\begin{aligned}
\lambda_{i,\mu} &= \sup\left\{\lambda : \mathrm{E}_{F_i}\left[\left(\frac{1-X}{1-\mu}\right)^\lambda\right] \le 1\right\} \\
&\ge \sup\left\{\lambda \in [1,2] : \mathrm{E}_{F_i}\left[\left(\frac{1-X}{1-\mu}\right)^\lambda\right] \le 1\right\} \\
&\ge \sup\left\{\lambda \in [1,2] : \mathrm{E}_{F_i}\left[(2-\lambda)\left(\frac{1-X}{1-\mu}\right)^1 + (\lambda-1)\left(\frac{1-X}{1-\mu}\right)^2\right] \le 1\right\} \\
&= \sup\left\{\lambda \in [1,2] : (2-\lambda)\frac{1-\mu_i}{1-\mu} + (\lambda-1)\frac{\sigma_i^2 + (1-\mu_i)^2}{(1-\mu)^2} \le 1\right\}. \qquad (36)
\end{aligned}$$

If $(\sigma_i^2 + (1-\mu_i)^2)(1-\mu)^{-2} \ge 1$, that is, if $\sigma_i^2 \ge (\mu_i - \mu)(2 - \mu_i - \mu)$ then λ satisfying

$$(2-\lambda)\frac{1-\mu_i}{1-\mu} + (\lambda-1)\frac{\sigma_i^2 + (1-\mu_i)^2}{(1-\mu)^2} = 1$$

is contained in $[1,2]$. Therefore we obtain (35) for this case by solving this equality. In the other case, the condition in (36) is satisfied by $\lambda = 2$ and we have $\lambda_{i,\mu} \ge 2$. ■

Appendix B. Proof of Lemma 7

We prove this lemma by the technique known as sensitivity analysis for optimization problems given below.

Proposition 17 (Fiacco, 1983, Corollary 3.4.3) *For a function $f(x,y) : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}$, let $f^*(y)$ be a local minimum of $f(x,y)$ in some neighborhood of x. Assume that there exists a point x^* such that*

- *$f(x,y)$ is twice continuously differentiable in some neighborhood of $(x^*, 0)$,*
- *$\Delta_x f(x,0)|_{x=x^*} = 0$, and*
- *$\Delta_x^2 f(x,0)|_{x=x^*}$ is positive definite.*

Then $\Delta_y f^*(y) = \Delta_y f(x,y)|_{x=x^*}$.

Proof From Lemma 6, for the case $\mathrm{E}_F[(1-\mu)/(1-X)] < 1$ we have $L_{\max}(F,\mu) = \mathrm{E}_F[\log((1-X)/(1-\mu))]$. Therefore,

$$\frac{\partial}{\partial \mu} L_{\max}(F,\nu) = \frac{1}{1-\mu} = \nu^*(F,\mu)$$

for $\mathrm{E}_F[(1-\mu)/(1-X)] < 1$ and

$$\lim_{\epsilon \downarrow 0} \frac{L_{\max}(F,\mu+\epsilon) - L_{\max}(F,\mu)}{\epsilon} = \frac{1}{1-\mu} = \nu^*(F,\mu)$$

for $\mathrm{E}_F[(1-\mu)/(1-X)] = 1$.

Now consider the case $\mathrm{E}_F[(1-\mu)/(1-X)] \geq 1$. In this case, $L_{\max}(F,\mu) = \max_{0\leq\nu\leq(1-\mu)^{-1}} L(\nu;F,\mu) = \max_\nu L(\nu;F,\mu)$ from $L'(0;F,\mu) = 0$, $L'((1-\mu)^{-1};F,\mu) \leq 0$ and the convexity of $L(\nu;F,\mu)$. For this unconstrained optimization problem it holds from Prop. 17 that

$$\frac{\mathrm{d}(\max_\nu L(\nu;F,\mu))}{\mathrm{d}\mu} = \left.\frac{\mathrm{d}L(\nu;F,\mu)}{\mathrm{d}\mu}\right|_{\nu=\nu^*} = \nu^*(F,\mu)\,.$$

Therefore, we obtain

$$\frac{\partial}{\partial\mu} L_{\max}(F,\mu) = \nu^*(F,\mu)$$

for $\mathrm{E}_F[(1-\mu)/(1-X)] > 1$ and

$$\lim_{\epsilon\uparrow 0} \frac{L_{\max}(F,\mu+\epsilon) - L_{\max}(F,\mu)}{\epsilon} = \nu^*(F,\mu)$$

for $\mathrm{E}_F[(1-\mu)/(1-X)] = 1$. ■

Appendix C. Proof of Theorem 5

In this appendix we show Theorem 5 on the refined (asymptotic) regret bound of IMED. We prove the theorem by the following lemma on a stopping time of a stochastic process.

Lemma 18 *Let $\{Y_i\}_{i=1,2,\cdots}$ be i.i.d. random variables such that $\mathrm{E}[Y_1] > 0$ and $\mathrm{E}[\mathrm{e}^{Y_1}] < \infty$.*
(i) *For $S_t = \sum_{i=1}^t Y_i$ and sufficiently large $M > 0$, the stopping time $\tau = \min\{t : S_t > M\}$ satisfies*

$$\mathrm{E}[\tau] \leq \frac{M + \log M}{\mathrm{E}[Y_1]} + \mathrm{O}(1)\,.$$

(ii) *Furthermore, if $\operatorname{ess\,sup} Y_i < \infty$, that is, the support of the distribution of Y_i is bounded from above then*

$$\mathrm{E}[\tau] \leq \frac{M}{\mathrm{E}[Y_1]} + \mathrm{O}(1)\,.$$

Proof (i) For any $A > 0$, define $Y_i' = Y_i \wedge A$ and $S_t' = \sum_{i=1}^t Y_i'$. For simplicity we also define $S_0' = S_0 = 0$. Since $S_t' \leq S_t$ always holds, $\tau' = \min\{t : S_t' > M\}$ satisfies $\tau \leq \tau'$.

Since $\tau_n' = n \wedge \tau'$ is a bounded stopping time, it holds from discrete Dynkin's formula (Meyn and Tweedie, 1992, Sect. 4.2) that

$$\begin{aligned}
\mathrm{E}[S'_{\tau_n'}] &= \mathrm{E}[S_0'] + \mathrm{E}\left[\sum_{i=1}^{\tau_n'} \mathrm{E}[S_i'|S_1', S_2', \cdots, S_{i-1}'] - S_{i-1}'\right] \\
&= \mathrm{E}\left[\sum_{i=1}^{\tau_n'} \mathrm{E}[Y_i']\right] \\
&= \mathrm{E}[Y_i']\mathrm{E}\left[\tau_n'\right]
\end{aligned}$$

and therefore

$$\mathrm{E}[\tau'_n] = \frac{\mathrm{E}[S'_{\tau'_n}]}{\mathrm{E}[Y'_1]} \le \frac{\mathrm{E}[S'_{\tau'_n-1} + A]}{\mathrm{E}[Y'_1]} \le \frac{M+A}{\mathrm{E}[Y'_1]}\,. \tag{37}$$

By defining $(x)_+ = 0 \vee x$, we can bound $\mathrm{E}[Y'_1]$ by

$$\begin{aligned}\mathrm{E}[Y'_1] &= \mathrm{E}[Y_1 - (Y_1 - A)_+] \\ &\ge \mathrm{E}[Y_1] - \frac{\mathrm{E}[\mathrm{e}^Y]}{\mathrm{e}^{A+1}}\,. \qquad (\text{by } (y-A)_+ \le \mathrm{e}^{y-(A+1)})\end{aligned} \tag{38}$$

Combining (37) with (38) and letting $A = \log((M+1)\mathrm{E}[\mathrm{e}^{Y_1}]/\mathrm{E}[Y_1]) - 1$, we have

$$\begin{aligned}\mathrm{E}[\tau'_n] &\le \frac{M+1}{M}\frac{M + \log\left(\frac{\mathrm{E}[\mathrm{e}^{Y_1}]}{\mathrm{E}[Y_1]}(M+1)\right) - 1}{\mathrm{E}[Y_1]} \\ &= \frac{M + \log M}{\mathrm{E}[Y_1]} + \mathrm{O}(1)\,.\end{aligned}$$

Finally we complete the proof by

$$\begin{aligned}\mathrm{E}[\tau] &\le \mathrm{E}[\tau'] \\ &= \mathrm{E}\left[\lim_{n\to\infty} \tau'_n\right] \\ &= \lim_{n\to\infty} \mathrm{E}[\tau'_n] \qquad (\text{by monotone convergence theorem}) \\ &= \frac{M + \log M}{\mathrm{E}[Y_1]} + \mathrm{O}(1)\,.\end{aligned}$$

(ii) In the case of $\operatorname{ess\,sup} Y_i < \infty$, we can directly evaluate τ instead of τ' and (37) is replaced with

$$\mathrm{E}[\tau] \le \frac{M + \operatorname{ess\,sup} Y_i}{\mathrm{E}[Y_1]} = \frac{M}{\mathrm{E}[Y_1]} + \mathrm{O}(1)\,. \qquad \blacksquare$$

Proof of Theorem 5 For simplicity we consider the case $K = 2$ and assume $\mu^* = \mu_1 > \mu_2$. We can prove the theorem for the case $K > 2$ in the same way (see Remark 2 below this proof).

First we define three constants independent of n by

$$\xi \equiv \frac{1}{2\log\frac{1-\mu_2}{1-\mu_1}} > 0 \tag{39}$$

$$\rho \equiv \frac{D_{\inf}(F_2, \mu_1)}{3} > 0$$

$$\mu' \equiv \max\left\{\mu_1 - \rho(1-\mu_1),\, \frac{\mu_1+\mu_2}{2}\right\} \in (\mu_2, \mu_1)\,. \tag{40}$$

We also define the following six events for sufficiently small $\delta > 0$

$$\begin{aligned}
A_l &\equiv \{J(l) = 2,\ T_2(l) \geq \xi \log n\}\,, \\
B_l^{(1)} &\equiv \{\hat{\mu}^*(l) \leq \mu'\}\,, \\
B_l^{(2)} &\equiv \{\mu' < \hat{\mu}^*(l) \leq \mu_1 - \delta\}\,, \\
B_l^{(3)} &\equiv \{\mu_1 - \delta < \hat{\mu}^*(l)\}\,, \\
C_l &\equiv \{\hat{\mu}_2(l) \leq \mu'\}\,, \\
D_l &\equiv \left\{D_{\inf}(\hat{F}_2(l), \mu_1) \geq D_{\inf}(F_2, \mu_1) - \rho\right\}\,.
\end{aligned}$$

Since the whole sample space is covered by

$$C_l^c \cup D_l^c \cup B_l^{(1)} \cup (B_l^{(2)} \cap C_l \cap D_l) \cup (B_l^{(3)} \cap C_l)\,,$$

we have

$$\begin{aligned}
T_2(n) &= \sum_{l=1}^{n} \mathbb{1}[J(l) = 2] \\
&\leq \xi \log n + \sum_{l=1}^{n} \mathbb{1}[A_l] \\
&\leq \sum_{l=1}^{n} \mathbb{1}[A_l \cap C_l^c] + \sum_{l=1}^{n} \mathbb{1}[A_l \cap D_l^c] + \sum_{l=1}^{n} \mathbb{1}\left[A_l \cap B_l^{(1)}\right] \\
&\quad + \sum_{l=1}^{n} \mathbb{1}\left[A_l \cap B_l^{(2)} \cap C_l \cap D_l\right] + \left(\xi \log n + \sum_{l=1}^{n} \mathbb{1}\left[A_l \cap B_l^{(3)} \cap C_l\right]\right). \qquad (41)
\end{aligned}$$

We bound expectations of these terms in the followings. The essential point is that the only events involving $B_l^{(2)}$ and $B_l^{(3)}$ depend on the small constant δ and the number of rounds of the other events can be bounded independently of δ. We can derive a tight bound for events $B_l^{(2)}$ and $B_l^{(3)}$ with respect to δ by considering these events under C_l and D_l, that is, under the condition that statistics $\hat{\mu}_2(l)$ and $D_{\inf}(\hat{F}_2(l), \mu_1)$ are not very far from the true expectation.

First we have[3]

$$\sum_{l=1}^{n} \mathbb{1}[A_l \cap C_l^c] \leq \sum_{t=\xi \log n}^{\infty} \mathbb{1}\left[\bigcup_{l=1}^{n} \{J(l) = 2,\ \hat{\mu}_{2,t} > \mu',\ T_2(l) = t\}\right] \qquad (42)$$

3. The summation $\sum_{t=\xi \log n}^{\infty}$ in (42) should be $\sum_{t=\lceil \xi \log n \rceil}^{\infty}$ to be precise. However we omit the rounding operations $\lceil \cdot \rceil$ and $\lfloor \cdot \rfloor$ in the proof of this theorem for simplicity since these do not affect the asymptotic analysis.

and therefore

$$\begin{aligned}
\mathrm{E}\left[\sum_{l=1}^{n} \mathbb{1}[A_l \cap C_l^c]\right] &\le \sum_{t=\xi\log n}^{\infty} P_{F_2}[\hat{\mu}_{2,t} > \mu'] \\
&\le \sum_{t=\xi\log n}^{\infty} \mathrm{e}^{-t\Lambda_2^*(\mu')} \qquad \text{(by (18) of Prop. 9)} \\
&= \frac{\mathrm{e}^{-(\xi\log n)\Lambda_2^*(\mu')}}{1-\mathrm{e}^{-\Lambda_2^*(\mu')}} \\
&= \mathrm{O}(\mathrm{e}^{-\mathrm{O}(\log n)}) \\
&= \mathrm{o}(1)\,. \qquad (43)
\end{aligned}$$

Second, we have

$$\sum_{l=1}^{n} \mathbb{1}[A_l \cap D_l^c] \le \sum_{t=\xi\log n}^{\infty} \mathbb{1}\left[\bigcup_{l=1}^{n} \left\{J(l)=2,\ D_{\inf}(\hat{F}_{2,t},\mu_1) < D_{\inf}(F_2,\mu_1)-\rho,\ T_2(l)=t\right\}\right].$$

From Prop. 10, its expectation is bounded as

$$\begin{aligned}
\mathrm{E}\left[\sum_{l=\xi\log n}^{n} \mathbb{1}[A_l \cap D_l^c]\right] &\le \sum_{t=\xi\log n}^{\infty} \mathrm{e}^{-t\tilde{\Lambda}_2^*(D_{\inf}(F_2,\mu_1)-\rho)} \\
&= \frac{\mathrm{e}^{-(\xi\log n)\tilde{\Lambda}_2^*(D_{\inf}(F_2,\mu_1)-\rho)}}{1-\mathrm{e}^{-\tilde{\Lambda}_2^*(D_{\inf}(F_2,\mu_1)-\rho)}} \\
&= \mathrm{o}(1)\,. \qquad (44)
\end{aligned}$$

Third, we have

$$\mathrm{E}\left[\sum_{l=\xi\log n}^{n} \mathbb{1}\left[A_l \cap B_l^{(1)}\right]\right] = \mathrm{O}(1) \qquad (45)$$

from Lemma 14 with $\mu := \mu'$ since μ' is a constant independent of δ and n.

Fourth, we have

$$\begin{aligned}
&\sum_{l=1}^{n} \mathbb{1}\left[A_l \cap B_l^{(2)} \cap C_l \cap D_l\right] \\
&\le \sum_{t_2=\xi\log n}^{\infty} \sum_{t_1=1}^{\infty} \mathbb{1}\left[\bigcup_{l=1}^{n} \{J(l)=2,\ T_1(l)=t_1,\ T_2(l)=t_2,\ B_l^{(2)} \cap C_l \cap D_l\}\right].
\end{aligned}$$

Note that $\{T_2(l)=t_2,\ B_l^{(2)} \cap D_l\}$ implies

$$\begin{aligned}
I_2(l) &\ge t_2 D_{\inf}(\hat{F}_2(l),\mu') \\
&\ge t_2\left(D_{\inf}(\hat{F}_2(l),\mu_1)-\rho\right) \qquad \text{(by (40) and Lemma 7)} \\
&\ge t_2\left(D_{\inf}(F_2,\mu_1)-2\rho\right) \qquad \text{(by definition of } D_l) \\
&= t_2\rho\,.
\end{aligned}$$

Furthermore, $J(l) = 2$ implies $I_2(l) \le I_1(l)$ and $\{T_1(l) = t_1,\, B_l^{(2)} \cap C_l\}$ implies $I_1(l) = \log t_1$. Combining them, we have

$$\begin{aligned}\sum_{l=1}^{n} \mathbb{1}\Big[A_l \cap B_l^{(2)} \cap C_l \cap D_l\Big] &\le \sum_{t_2=\xi\log n}^{\infty} \sum_{t_1=1}^{\infty} \mathbb{1}[\rho t_2 \le \log t_1,\ \hat{\mu}_{1,t_1} \le \mu_1 - \delta] \\ &= \sum_{t_2=\xi\log n}^{\infty} \sum_{t_1=\mathrm{e}^{\rho t_2}}^{\infty} \mathbb{1}[\hat{\mu}_{1,t_1} \le \mu_1 - \delta] \qquad (46)\end{aligned}$$

and therefore

$$\begin{aligned}\mathrm{E}\left[\sum_{l=1}^{n} \mathbb{1}\Big[A_l \cap B_l^{(2)} \cap C_l \cap D_l\Big]\right] &\le \sum_{t_2=\xi\log n}^{\infty} \sum_{t_1=\mathrm{e}^{\rho t_2}}^{\infty} P_{F_1}\left[\hat{\mu}_{1,t_1} \le \mu_1 - \delta\right] \\ &\le \sum_{t_2=\xi\log n}^{\infty} \frac{\mathrm{e}^{-\mathrm{e}^{\rho t_2}\Lambda_1^*(\mu_1-\delta)}}{1-\mathrm{e}^{-\Lambda_1^*(\mu_1-\delta)}} \qquad \text{(by (17) of Prop. 9)} \\ &\le \sum_{t_2=\xi\log n}^{\infty} \frac{\mathrm{e}^{-(\frac{(\rho t_2)^3}{3}+\rho t_2)\Lambda_1^*(\mu_1-\delta)}}{1-\mathrm{e}^{-\Lambda_1^*(\mu_1-\delta)}} \\ &\qquad\qquad \text{(by } \mathrm{e}^x \ge \tfrac{x^3}{3} + x \text{ for } x \ge 0) \\ &\le \sum_{t_2=\xi\log n}^{\infty} \frac{\mathrm{e}^{-(\frac{(\rho\xi\log n)^3}{3}+\rho t_2)\Lambda_1^*(\mu_1-\delta)}}{1-\mathrm{e}^{-\Lambda_1^*(\mu_1-\delta)}} \\ &= \frac{\mathrm{e}^{-(\frac{(\rho\xi\log n)^3}{3}+\rho\xi\log n)\Lambda_1^*(\mu_1-\delta)}}{(1-\mathrm{e}^{-\Lambda_1^*(\mu_1-\delta)})(1-\mathrm{e}^{-\rho\Lambda_1^*(\mu_1-\delta)})} \\ &= \frac{\mathrm{e}^{-\mathrm{O}(\delta^2(\log n)^3)}}{\mathrm{O}(\delta^4)}\,. \qquad (47)\end{aligned}$$

Finally we evaluate two terms

$$\xi\log n + \sum_{l=1}^{n} \mathbb{1}\Big[A_l \cap B_l^{(3)} \cap C_l\Big] = \xi\log n + \sum_{t=\xi\log n}^{n} \mathbb{1}\left[\bigcup_{l=1}^{n}\{J(l) = 2,\ T_2(l) = t,\ B_l^{(3)} \cap C_l\}\right]$$

in (41). Here note that $\{T_2(l) = t \ge \xi\log n,\ B_l^{(3)}\}$ implies

$$\begin{aligned}I_2(l) &\ge t D_{\inf}(\hat{F}_2, \mu_1 - \delta) + \log t \\ &\ge t\left(D_{\inf}(\hat{F}_2, \mu_1) - \frac{\delta}{1-\mu_1}\right) + \log(\xi\log n) \qquad \text{(by Lemma 7)}\end{aligned}$$

and $\{J(l) = 2,\ B_l^{(3)} \cap C_l\}$ implies $I_2(l) \le I_1(l) = \log T_1(l) \le \log n$. As a result, we have

$$\xi\log n + \sum_{l=1}^{n} \mathbb{1}\Big[A_l \cap B_l^{(3)} \cap C_l\Big]$$

$$\leq \xi \log n + \sum_{t=\xi \log n}^{\infty} \mathbb{1}\left[t \left(D_{\inf}(\hat{F}_2, \mu_1) - \frac{\delta}{1-\mu_1} \right) \leq \log n - \log(\xi \log n) \right]$$
$$= \sum_{t=1}^{\infty} \mathbb{1}\left[t \left(D_{\inf}(\hat{F}_{2,t}, \mu_1) - \frac{\delta}{1-\mu_1} \right) \leq \log n - \log(\xi \log n) \right]$$
$$+ \sum_{t=1}^{\xi \log n} \mathbb{1}\left[t \left(D_{\inf}(\hat{F}_{2,t}, \mu_1) - \frac{\delta}{1-\mu_1} \right) > \log n - \log(\xi \log n) \right] . \tag{48}$$

The expectation of the second term of (48) can be evaluated as

$$\mathrm{E}\left[\sum_{t=1}^{\xi \log n} \mathbb{1}\left[t \left(D_{\inf}(\hat{F}_{2,t}, \mu_1) - \frac{\delta}{1-\mu_1} \right) > \log n - \log(\xi \log n) \right] \right]$$
$$\leq \sum_{t=1}^{\xi \log n} P_{F_2}\left[D_{\inf}(\hat{F}_{2,t}, \mu_1) > \frac{\log n - \log(\xi \log n)}{\xi \log n} \right]$$
$$= \sum_{t=1}^{\xi \log n} P_{F_2}\left[D_{\inf}(\hat{F}_{2,t}, \mu_1) > \frac{1}{\xi} - \mathrm{o}(1) \right]$$
$$= \sum_{t=1}^{\xi \log n} P_{F_2}\left[D_{\inf}(\hat{F}_{2,t}, \mu_1) > 2 \log \frac{1-\mu_2}{1-\mu_1} - \mathrm{o}(1) \right] \qquad \text{(by (39))}$$
$$= \mathrm{O}(1) . \qquad \text{(by Prop. 12)} \tag{49}$$

Putting (41) and (43)–(49) together, we have

$$\mathrm{E}[T_2(n)] \leq \mathrm{E}\left[\sum_{t=1}^{\infty} \mathbb{1}\left[t \left(D_{\inf}(\hat{F}_2, \mu_1) - \frac{\delta}{1-\mu_1} \right) \leq \log n - \log(\xi \log n) \right] \right]$$
$$+ \frac{\mathrm{e}^{-\mathrm{O}(\delta^2 (\log n)^3)}}{\mathrm{O}(\delta^4)} + \mathrm{O}(1) . \tag{50}$$

Let $Y_t = \log(1 - (X_{2,t} - \mu_1)\nu_2^*) - \delta/(1-\mu_1)$ and define a stochastic process $\{S_t\}_{t=1,2,\cdots}$ by $S_t = \sum_{l=1}^{t} Y_l$. For a stopping time $\tau = \min\{t : S_t > \log n - \log(\xi \log n)\}$, the first term of (50) is bounded by

$$\mathrm{E}\left[\sum_{t=1}^{\infty} \mathbb{1}\left[t \left(D_{\inf}(\hat{F}_{2,t}, \mu_1) - \frac{\delta}{1-\mu_1} \right) \leq \log n - \log(\xi \log n) \right] \right]$$
$$\leq \mathrm{E}\left[\sum_{t=1}^{\infty} \mathbb{1}[S_t \leq \log n - \log(\xi \log n)] \right]$$
$$= \mathrm{E}\left[(\tau - 1) + \sum_{m=\tau+1}^{n} \mathbb{1}\left[S_\tau + \sum_{l=\tau+1}^{m} Y_l \leq \log n - \log(\xi \log n) \right] \right]$$
$$\leq \mathrm{E}[\tau] + \mathrm{E}\left[\sum_{m=\tau+1}^{n} \mathbb{1}\left[\sum_{l=\tau+1}^{m} Y_l \leq 0 \right] \right]$$

$$= \mathrm{E}[\tau] + \mathrm{E}\left[\mathrm{E}\left[\sum_{m=\tau+1}^{n} \mathbb{1}\left[\sum_{l=\tau+1}^{m} Y_l \le 0\right] \middle| \tau\right]\right]$$
$$= \mathrm{E}[\tau] + \mathrm{E}\left[\sum_{m=\tau+1}^{n} P_{F_2}\left[\sum_{l=\tau+1}^{m} Y_l \le 0 \middle| \tau\right]\right]. \tag{51}$$

Note that $\mathrm{E}[Y_t] = D_{\inf}(F_2, \mu_1) - \delta/(1-\mu_1)$ and $\mathrm{E}[\mathrm{e}^{Y_t}] = \mathrm{e}^{-\delta/(1-\mu_1)}(1-(\mu_2-\mu_1)\nu_i^*) < \infty$. Then we obtain from (i) of Lemma 18 that

$$\begin{aligned}\mathrm{E}[\tau] &\le \frac{\log n - \log(\xi \log n) + \log(\log n - \log(\xi \log n))}{D_{\inf}(F_2, \mu_1) - \frac{\delta}{1-\mu_1}} + \mathrm{O}(1) \\ &= \frac{\log n}{D_{\inf}(F_2, \mu_1) - \frac{\delta}{1-\mu_1}} + \mathrm{O}(1) \\ &= \frac{\log n}{D_{\inf}(F_2, \mu_1)} + \mathrm{O}(\delta \log n) + \mathrm{O}(1)\,. \end{aligned} \tag{52}$$

On the other hand, from Cramér's theorem we obtain

$$\begin{aligned}&\mathrm{E}\left[\sum_{m=\tau+1}^{n} P_{F_2}\left[\sum_{l=\tau+1}^{m} Y_l \le 0 \middle| \tau\right]\right] \\ &= \mathrm{E}\left[\sum_{m=\tau+1}^{n} P_{F_2}\left[\frac{1}{m-\tau}\sum_{l=\tau+1}^{m} \log(1-(X_{2,l}-\mu_1)\nu_2^*) \le \frac{\delta}{1-\mu_1} \middle| \tau\right]\right] \\ &\le \mathrm{E}\left[\sum_{m=\tau+1}^{n} \mathrm{e}^{-(m-\tau)\tilde{\Lambda}_2^*(\frac{\delta}{1-\mu_1})}\right] \qquad \text{(by Prop. 9 and definition of } \tilde{\Lambda}_2^* \text{ in (6))} \\ &\le \frac{1}{1-\mathrm{e}^{-\tilde{\Lambda}_2^*(\frac{\delta}{1-\mu_1})}} \\ &= \mathrm{O}(1)\,. \qquad \text{(by Lemma 15)} \end{aligned} \tag{53}$$

By combining (51)–(53) with (50) we have

$$\mathrm{E}[T_2(n)] \le \frac{\log n}{D_{\inf}(F_2, \mu_1)} + \mathrm{O}(\delta \log n) + \frac{\mathrm{e}^{-\mathrm{O}(\delta^2 (\log n)^3)}}{\mathrm{O}(\delta^4)} + \mathrm{O}(1)\,.$$

We obtain (i) of Theorem 5 by letting $\delta = \mathrm{O}((\log n)^{-1})$.

In the case that each arm has a bounded support we can apply (ii) of Lemma 18. As a result, (52) is replaced with

$$\begin{aligned}\mathrm{E}[\tau] &\le \frac{\log n - \log(\xi \log n)}{D_{\inf}(F_2, \mu_1) - \frac{\delta}{1-\mu_1}} + \mathrm{O}(1) \\ &= \frac{\log n}{D_{\inf}(F_2, \mu_1)} + \mathrm{O}(\delta \log n) - \mathrm{O}(\log\log n) \end{aligned}$$

and we obtain (ii) of Theorem 5 by this replacement. ■

Remark 2 The proof for $K > 2$ is almost the same as the case $K = 2$. The only different point is the evaluation around (46), wherein the pair $(T_1(l), T_2(l))$ is considered. For $K > 3$ we can proceed the evaluation in the same way by taking the summation over contributions of all pairs $(T_j(l), T_i(l))$, $j \in \mathcal{I}_{\rm opt}, i \neq j$.

References

Rajeev Agrawal. The continuum-armed bandit problem. *SIAM Journal on Control and Optimization*, 33(6):1926–1951, 1995.

Shipra Agrawal and Navin Goyal. Further optimal regret bounds for Thompson sampling. In *Proceedings of AISTATS 2010*, volume 31, pages 99–107, 2013.

Jean-Yves Audibert, Rémi Munos, and Csaba Szepesvári. Exploration-exploitation tradeoff using variance estimates in multi-armed bandits. *Theoretical Computer Science*, 410: 1876–1902, April 2009.

Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47:235–256, 2002a.

Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2002b.

Jonathan M. Borwein and Adrian S. Lewis. Partially-finite programming in L_1 and the existence of maximum entropy estimates. *SIAM Journal on Optimization*, 3(2):248–267, 1993.

Sébastien Bubeck, Nicolò Cesa-Bianchi, and Gábor Lugosi. Bandits with heavy tail. *arXiv*, 2012. URL `http://arxiv.org/abs/1209.1727`.

Apostolos N. Burnetas and Michael N. Katehakis. Optimal adaptive policies for sequential allocation problems. *Advances in Applied Mathematics*, 17(2):122–142, 1996.

Olivier Cappé, Aurélien Garivier, Odalric-Ambrym Maillard, Rémi Munos, and Gilles Stoltz. Kullback-Leibler upper confidence bounds for optimal sequential allocation. *Annals of Statistics*, 41(3):1516–1541, 2013.

Oliver Chapelle and Lihong Li. An empirical evaluation of Thompson sampling. In *Proceedings of NIPS 2011*, volume 24, pages 1252–1260, Granada, Spain, 2012.

Yuan S. Chow and Tze L. Lai. Some one-sided theorems on the tail distribution of sample sums with applications to the last time and largest excess of boundary crossings. *Transactions of the American Mathematical Society*, 208:51–72, 1975.

Amir Dembo and Ofer Zeitouni. *Large Deviations Techniques and Applications*, volume 38 of *Applications of Mathematics*. Springer-Verlag, New York, second edition, 1998.

Anthony V. Fiacco. *Introduction to Sensitivity and Stability Analysis in Nonlinear Programming*. Academic Press, New York, 1983.

Aurelien Garivier and Olivier Cappé. The KL-UCB algorithm for bounded stochastic bandits and beyond. In *Proceedings of COLT 2011*, Budapest, Hungary, 2011.

John C. Gittins. *Multi-armed Bandit Allocation Indices*. Wiley-Interscience Series in Systems and Optimization. John Wiley & Sons, Chichester, 1989.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

Junya Honda and Akimichi Takemura. An asymptotically optimal bandit algorithm for bounded support models. In *Proceedings of COLT 2010*, pages 67–79, Haifa, Israel, 2010.

Junya Honda and Akimichi Takemura. An asymptotically optimal policy for finite support models in the multiarmed bandit problem. *Machine Learning*, 85(3):361–391, 2011.

Junya Honda and Akimichi Takemura. Stochastic bandit based on empirical moments. In *Proceedings of AISTATS 2012*, pages 529–537, Canary Islands, Spain, 2012.

S Ito, Y Liu, and K. L. Teo. A dual parametrization method for convex semi-infinite programming. *Annals of Operations Research*, 98(1-4):189–213, 2000.

Samuel Karlin and William J. Studden. *Tchebycheff Systems, with Applications in Analysis and Statistics*. Interscience Publishers New York, 1966.

Emilie Kaufmann. *Analyse de stratégies bayésiennes et fréquentistes pour l'allocation séquentielle de ressources*. PhD thesis, TELECOM ParisTech, 2014.

Emilie Kaufmann, Olivier Cappé, and Aurélien Garivier. On Bayesian upper confidence bounds for bandit problems. In *Proceedings of AISTATS 2012*, pages 592–600, 2012a.

Emilie Kaufmann, Nathaniel Korda, and Rémi Munos. Thompson sampling: an asymptotically optimal finite-time analysis. In *Proceedings of ALT 2012*, pages 199–213, Berlin, Heidelberg, 2012b. Springer-Verlag.

Nathaniel Korda, Emilie Kaufmann, and Remi Munos. Thompson sampling for 1-dimensional exponential family bandits. In *Proceedings of NIPS 2013*, Lake Tahoe, NV, USA, 2013.

Balachander Krishnamurthy, Craig Wills, and Yin Zhang. On the use and performance of content distribution networks. In *Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement*, pages 169–182, New York, USA, 2001.

Tze L. Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6:4–22, 1985.

Keqin Liu and Qing Zhao. Multi-armed bandit problems with heavy-tailed reward distributions. In *2011 49th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 485–492. IEEE, 2011.

Sean P. Meyn and R. L. Tweedie. Stability of Markovian processes I: Criteria for discrete-time chains. *Advances in Applied Probability*, 24:542–574, 1992.

Herbert Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58(5):527–35, 1952.

Daniel Russo and Benjamin V. Roy. Learning to optimize via posterior sampling. *arXiv*, 2013. URL `http://arxiv.org/abs/1301.2609`.

William R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25:285–294, 1933.

Joannès Vermorel and Mehryar Mohri. Multi-armed bandit algorithms and empirical evaluation. In *Proceedings of ECML 2005*, pages 437–448, Porto, Portugal, 2005. Springer.

Journal of Machine Learning Research 16 (2015) 3757-3811 Submitted 5/14; Revised 3/15; Published 12/15

Condition for Perfect Dimensionality Recovery by Variational Bayesian PCA*

Shinichi Nakajima NAKAJIMA@TU-BERLIN.DE
Berlin Big Data Center
Technische Universität Berlin
Berlin 10587 Germany

Ryota Tomioka TOMIOKA@TTIC.EDU
Toyota Technological Institute at Chicago
Chicago, IL 60637 USA

Masashi Sugiyama SUGI@K.U-TOKYO.AC.JP
Department of Complexity Science and Engineering
The University of Tokyo
Tokyo 113-0033 Japan

S. Derin Babacan DBABACAN@GMAIL.COM
Google Inc.
Mountain View, CA 94043 USA

Editor: David Barber

Abstract

Having shown its good performance in many applications, variational Bayesian (VB) learning is known to be one of the best tractable approximations to Bayesian learning. However, its performance was not well understood theoretically. In this paper, we clarify the behavior of VB learning in probabilistic PCA (or fully-observed matrix factorization). More specifically, we establish a necessary and sufficient condition for perfect dimensionality (or rank) recovery in the large-scale limit when the matrix size goes to infinity. Our result theoretically guarantees the performance of VB-PCA. At the same time, it also reveals the conservative nature of VB learning—it offers a low false positive rate at the expense of low sensitivity. By contrasting with an alternative dimensionality selection method, we characterize VB learning in PCA. In our analysis, we obtain bounds of the noise variance estimator, and a new and simple analytic-form solution for the other parameters, which themselves are useful for implementation of VB-PCA.

Keywords: variational Bayesian learning, matrix factorization, principal component analysis, automatic relevance determination, perfect dimensionality recovery

1. Introduction

Variational Bayesian (VB) learning (Attias, 1999; Bishop, 2006) was proposed as a computationally efficient approximation to Bayesian learning. The key idea is to find the closest distribution to the Bayes posterior in a restricted function space, where the expectation—an often intractable operation in Bayesian learning—can be easily performed. VB learning

* This paper is an extended version of our earlier conference paper (Nakajima et al., 2012).

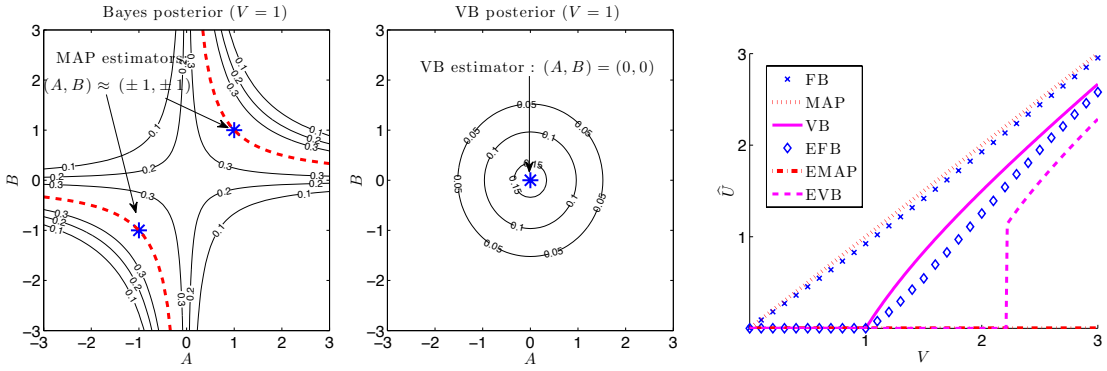


Figure 1: Dissimilarities between VB and rigorous Bayesian learning. (Left and Center) The Bayes posterior and the VB posterior of the 1×1 MF model $V = BA + \mathcal{E}$ with almost flat prior, when $V = 1$ is observed ($\mathcal{E}$ is Gaussian noise). VB approximates the Bayes posterior having two modes by an origin-centered Gaussian, which induces sparsity. (Right) Behavior of estimators of $U = BA$, given the observation V. The VB estimator (the magenta solid curve) is zero when $V \leq 1$, which indicates *exact* sparsity. On the other hand, FB (fully-Bayesian or rigorous Bayesian learning; blue crosses) shows no sign of sparsity. All graphs are quoted from Nakajima and Sugiyama (2011).

has been applied to many applications, and its good performance has been experimentally shown (Bishop, 1999a; Bishop and Tipping, 2000; Ghahramani and Beal, 2001; Jaakkola and Jordan, 2000; Blei et al., 2003; Sato et al., 2004; Lim and Teh, 2007; Seeger, 2009; Ilin and Raiko, 2010). Typically, the restriction is imposed as a factorized form of posterior, under which a tractable iterative algorithm is derived.

Although the VB algorithm is simple and efficient, it solves a non-convex optimization problem, which makes theoretical analysis difficult. An exceptional case is the matrix factorization (MF) model (Bishop, 1999a; Lim and Teh, 2007; Ilin and Raiko, 2010; Salakhutdinov and Mnih, 2008) with fully-observed matrices, in which the global VB solution has been analytically obtained (Nakajima et al., 2013b), and some properties have been theoretically revealed (Nakajima and Sugiyama, 2011). These works also posed thought-provoking relations between VB and rigorous Bayesian learning: The VB posterior is actually quite different from the true Bayes posterior (compare the left and the middle graphs in Figure 1), and VB induces sparsity in its solution but such sparsity is hardly observed in rigorous Bayesian learning (see the right graph in Fig. 1). Actually, Mackay (2001) has discussed the sparsity of VB as an artifact by showing *inappropriate* model pruning in mixture models. These facts might deprive the justification of VB based solely on the fact that it is one of the best tractable approximations to Bayesian learning.

The goal of this paper is to provide direct justification for VB learning. Focusing on the probabilistic PCA (Tipping and Bishop, 1999; Roweis and Ghahramani, 1999; Bishop, 1999a), an instance of fully-observed MF, we give a theoretical guarantee for the performance of VB learning. Our starting point is the global analytic solution derived by Nakajima

et al. (2013b). After describing our formulation in Section 2, we conduct the following three steps:

1. We derive a new and simple analytic-form of the global VB solution in Section 3.

   The analytic-form solution derived in Nakajima et al. (2013b) is expressed with a solution of a *quartic* equation, which obstructs further analysis. In this paper, we derive an alternative form, which consists of simple algebra.

2. We obtain a simple form of the objective function for noise variance estimation in Section 4.

   The previous analyses in Nakajima and Sugiyama (2011) and in Nakajima et al. (2013b) assumed that the noise variance is a given constant. In this paper, we assume that the noise variance is also estimated from observation, and derive an objective function, of which the minimizer gives the noise variance estimator. We also derive bounds of the rank estimator and the noise variance estimator.

3. We establish a necessary and sufficient condition for perfect dimensionality recovery in Section 5.

   Combining the results obtained in the former two steps with random matrix theory (Marčenko and Pastur, 1967; Wachter, 1978; Johnstone, 2001; Hoyle and Rattray, 2004; Baik and Silverstein, 2006), we establish a necessary and sufficient condition that VB-PCA perfectly recovers the true dimensionality in the *large-scale limit* when the matrix size goes to infinity.

To the best of our knowledge, this is the first theoretical result that guarantees the performance of VB learning. To give more insight into practical situations, we also derive a sufficient condition for perfect recovery, which approximately holds for moderate-sized matrices. It is worth noting that, although the objective function minimized for noise variance estimation is non-convex and possibly multimodal in general, only a local search algorithm is required for perfect recovery.

Section 6 is devoted to discussion on a few topics. First, we propose a simple implementation of VB-PCA, based on the new analytic-form solution and the bounds of the noise variance estimator, which are obtained in our analysis. After that, we consider the behavior of VB learning in more detail. Our result theoretically guarantees the performance of VB-PCA. At the same time, it also reveals the conservative nature of VB learning—it offers a low false positive rate at the expense of low sensitivity, due to which VB-PCA does not behave *optimally* in the large-scale limit. By contrasting with an alternative dimensionality selection method, called the *overlap* (OL) method (Hoyle, 2008), we characterize VB learning in PCA.

Section 7 concludes, and Appendix provides all technical details.

2. Formulation

In this section, we formulate variational Bayesian learning in the matrix factorization model.

2.1 Probabilistic Matrix Factorization

Assume that we observed a matrix $\boldsymbol{V} \in \mathbb{R}^{L \times M}$, which is the sum of a target matrix $\boldsymbol{U} \in \mathbb{R}^{L \times M}$ and a noise matrix $\boldsymbol{\mathcal{E}} \in \mathbb{R}^{L \times M}$:

$$\boldsymbol{V} = \boldsymbol{U} + \boldsymbol{\mathcal{E}}.$$

In the *matrix factorization* (MF) model, the target matrix is assumed to be low rank, and therefore can be factorized as

$$\boldsymbol{U} = \boldsymbol{B}\boldsymbol{A}^\top,$$

where $\boldsymbol{A} \in \mathbb{R}^{M \times H}$, $\boldsymbol{B} \in \mathbb{R}^{L \times H}$ for $H \leq \min(L, M)$, and $\top$ denotes the transpose of a matrix or vector. Here, the rank of $\boldsymbol{U}$ is upper-bounded by H.

In this paper, we consider the probabilistic MF model (Salakhutdinov and Mnih, 2008), where the observation noise $\boldsymbol{\mathcal{E}}$ and the priors of $\boldsymbol{A}$ and $\boldsymbol{B}$ are assumed to be Gaussian:

$$p(\boldsymbol{V}|\boldsymbol{A}, \boldsymbol{B}) \propto \exp\left(-\frac{1}{2\sigma^2}\|\boldsymbol{V} - \boldsymbol{B}\boldsymbol{A}^\top\|_{\mathrm{Fro}}^2\right), \tag{1}$$

$$p(\boldsymbol{A}) \propto \exp\left(-\frac{1}{2}\mathrm{tr}\left(\boldsymbol{A}\boldsymbol{C}_A^{-1}\boldsymbol{A}^\top\right)\right), \tag{2}$$

$$p(\boldsymbol{B}) \propto \exp\left(-\frac{1}{2}\mathrm{tr}\left(\boldsymbol{B}\boldsymbol{C}_B^{-1}\boldsymbol{B}^\top\right)\right). \tag{3}$$

Here, we denote by $\|\cdot\|_{\mathrm{Fro}}$ the Frobenius norm, and by $\mathrm{tr}(\cdot)$ the trace of a matrix. Throughout the paper, we assume that

$$L \leq M. \tag{4}$$

If $L > M$, we may simply re-define the transpose $\boldsymbol{V}^\top$ as $\boldsymbol{V}$ so that $L \leq M$ holds. Therefore, the assumption (4) does not impose any restriction. We assume that the prior covariance matrices $\boldsymbol{C}_A$ and $\boldsymbol{C}_B$ are diagonal and positive definite, i.e.,

$$\boldsymbol{C}_A = \mathbf{diag}(c_{a_1}^2, \ldots, c_{a_H}^2),$$
$$\boldsymbol{C}_B = \mathbf{diag}(c_{b_1}^2, \ldots, c_{b_H}^2),$$

for $c_{a_h}, c_{b_h} > 0, h = 1, \ldots, H$. Without loss of generality, we assume that the diagonal entries of the product $\boldsymbol{C}_A\boldsymbol{C}_B$ are arranged in non-increasing order, i.e., $c_{a_h}c_{b_h} \geq c_{a_{h'}}c_{b_{h'}}$ for any pair $h < h'$. We denote a column vector of a matrix by a bold lowercase letter, i.e.,

$$\boldsymbol{A} = (\boldsymbol{a}_1, \ldots, \boldsymbol{a}_H) \in \mathbb{R}^{M \times H},$$
$$\boldsymbol{B} = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_H) \in \mathbb{R}^{L \times H}.$$

2.2 Variational Bayesian Approximation

The Bayes posterior is given by

$$p(\boldsymbol{A}, \boldsymbol{B}|\boldsymbol{V}) = \frac{p(\boldsymbol{V}|\boldsymbol{A}, \boldsymbol{B})p(\boldsymbol{A})p(\boldsymbol{B})}{p(\boldsymbol{V})}, \tag{5}$$

where $p(\boldsymbol{V}) = \langle p(\boldsymbol{V}|\boldsymbol{A},\boldsymbol{B})\rangle_{p(\boldsymbol{A})p(\boldsymbol{B})}$. Here, $\langle\cdot\rangle_p$ denotes the expectation over the distribution p. Since this expectation is intractable, we need to approximate the Bayes posterior.

Let $r(\boldsymbol{A},\boldsymbol{B})$, or r for short, be a trial distribution. The following functional with respect to r is called the free energy:

$$F(r) = \left\langle \log \frac{r(\boldsymbol{A},\boldsymbol{B})}{p(\boldsymbol{V}|\boldsymbol{A},\boldsymbol{B})p(\boldsymbol{A})p(\boldsymbol{B})} \right\rangle_{r(\boldsymbol{A},\boldsymbol{B})} = \left\langle \log \frac{r(\boldsymbol{A},\boldsymbol{B})}{p(\boldsymbol{A},\boldsymbol{B}|\boldsymbol{V})} \right\rangle_{r(\boldsymbol{A},\boldsymbol{B})} - \log p(\boldsymbol{V}). \tag{6}$$

In the last equation, the first term is the Kullback-Leibler (KL) divergence from the trial distribution to the Bayes posterior (5), and the second term is constant. Therefore, minimizing the free energy amounts to finding a distribution closest to the Bayes posterior in the sense of the KL divergence. A general approach to Bayesian approximate inference is to find the minimizer of the free energy (6) with respect to r in some restricted function space.

In the VB approximation, the independence between the entangled parameter matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ is assumed:

$$r(\boldsymbol{A},\boldsymbol{B}) = r(\boldsymbol{A})r(\boldsymbol{B}). \tag{7}$$

Under this constraint, an iterative algorithm for minimizing the free energy (6) was derived (Bishop, 1999a; Lim and Teh, 2007). Let $\widehat{r}$ be the obtained minimizer. We define the MF solution by the mean of the target matrix $\boldsymbol{U}$:

$$\widehat{\boldsymbol{U}} = \left\langle \boldsymbol{B}\boldsymbol{A}^\top \right\rangle_{\widehat{r}(\boldsymbol{A},\boldsymbol{B})}.$$

The MF model has hyperparameters $(\boldsymbol{C}_A, \boldsymbol{C}_B)$ in the priors (2) and (3). By manually choosing them, we can control regularization and sparsity of the solution (e.g., the PCA dimension in our setting). A popular way to set the hyperparameter in the Bayesian framework is again based on the minimization of the free energy (6):

$$(\widehat{\boldsymbol{C}}_A, \widehat{\boldsymbol{C}}_B) = \operatorname*{argmin}_{\boldsymbol{C}_A,\boldsymbol{C}_B} \left(\min_r F(r; \boldsymbol{C}_A, \boldsymbol{C}_B|\boldsymbol{V}) \right).$$

We refer to this method as an empirical VB (EVB) method. When the noise variance σ^2 is unknown, it can also be estimated as

$$\widehat{\sigma}^2 = \operatorname*{argmin}_{\sigma^2} \left(\min_{r,\boldsymbol{C}_A,\boldsymbol{C}_B} F(r; \boldsymbol{C}_A, \boldsymbol{C}_B, \sigma^2|\boldsymbol{V}) \right).$$

3. Simple Analytic-Form Solution

Recently, an analytic-form of the global VB, as well as EVB, solution for MF has been derived (Nakajima et al., 2013b), which enables us to reach the global solution easily. However, the form involves a solution of a *quartic* equation, which obstructs further analysis. In this section, we derive a simple alternative form of the global VB, as well as EVB, solution, which facilitates subsequent analysis.

3.1 VB Solution

Let

$$\boldsymbol{V} = \sum_{h=1}^{H} \gamma_h \boldsymbol{\omega}_{b_h} \boldsymbol{\omega}_{a_h}^\top$$

be the singular value decomposition (SVD) of $\boldsymbol{V}$, where γ_h (≥ 0) is the h-th largest singular value, and $\boldsymbol{\omega}_{a_h}$ and $\boldsymbol{\omega}_{b_h}$ are the associated right and left singular vectors. We denote by $\mathcal{N}_d(\cdot; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ the d-dimensional Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$, by $\boldsymbol{I}_d$ the d-dimensional identity matrix, and by $\mathbb{R}_{++}$ the set of the positive real numbers.

Under the independence assumption (7), it is easily shown that the VB posterior has the Gaussian form:

$$r(\boldsymbol{A}, \boldsymbol{B}) \propto \exp\left(-\frac{\mathrm{tr}\left((\boldsymbol{A}-\widehat{\boldsymbol{A}})\boldsymbol{\Sigma}_A^{-1}(\boldsymbol{A}-\widehat{\boldsymbol{A}})^\top\right)}{2}\right) \exp\left(-\frac{\mathrm{tr}\left((\boldsymbol{B}-\widehat{\boldsymbol{B}})\boldsymbol{\Sigma}_B^{-1}(\boldsymbol{B}-\widehat{\boldsymbol{B}})^\top\right)}{2}\right)$$

with the means $\widehat{\boldsymbol{A}}, \widehat{\boldsymbol{B}}$ and the covariances $\boldsymbol{\Sigma}_A, \boldsymbol{\Sigma}_B$ minimizing the free energy (6), which is explicitly written as

$$\begin{aligned} 2F = LM\log(2\pi\sigma^2) + \frac{\left\|\boldsymbol{V} - \widehat{\boldsymbol{B}}\widehat{\boldsymbol{A}}^\top\right\|^2}{\sigma^2} + M\log\frac{|\boldsymbol{C}_A|}{|\boldsymbol{\Sigma}_A|} + L\log\frac{|\boldsymbol{C}_B|}{|\boldsymbol{\Sigma}_B|} - (L+M)H \\ + \mathrm{tr}\left(\boldsymbol{C}_A^{-1}\left(\widehat{\boldsymbol{A}}^\top\widehat{\boldsymbol{A}} + M\boldsymbol{\Sigma}_A\right)\right) + \mathrm{tr}\left(\boldsymbol{C}_B^{-1}\left(\widehat{\boldsymbol{B}}^\top\widehat{\boldsymbol{B}} + L\boldsymbol{\Sigma}_B\right)\right) \\ + \frac{\mathrm{tr}\left(-\widehat{\boldsymbol{A}}^\top\widehat{\boldsymbol{A}}\widehat{\boldsymbol{B}}^\top\widehat{\boldsymbol{B}} + \left(\widehat{\boldsymbol{A}}^\top\widehat{\boldsymbol{A}} + M\boldsymbol{\Sigma}_A\right)\left(\widehat{\boldsymbol{B}}^\top\widehat{\boldsymbol{B}} + L\boldsymbol{\Sigma}_{\boldsymbol{B}}\right)\right)}{\sigma^2}. \end{aligned} \tag{8}$$

Here $|\cdot|$ denotes the determinant of a matrix. The derivatives of the free energy (8) give the following stationary condition, which is used for constructing an iterative local search algorithm:

$$\widehat{\boldsymbol{A}} = \boldsymbol{V}^\top\widehat{\boldsymbol{B}}\frac{\boldsymbol{\Sigma}_A}{\sigma^2}, \qquad \boldsymbol{\Sigma}_A = \sigma^2\left(\widehat{\boldsymbol{B}}^\top\widehat{\boldsymbol{B}} + L\boldsymbol{\Sigma}_B + \sigma^2\boldsymbol{C}_A^{-1}\right)^{-1}, \tag{9}$$

$$\widehat{\boldsymbol{B}} = \boldsymbol{V}\widehat{\boldsymbol{A}}\frac{\boldsymbol{\Sigma}_B}{\sigma^2}, \qquad \boldsymbol{\Sigma}_B = \sigma^2\left(\widehat{\boldsymbol{A}}^\top\widehat{\boldsymbol{A}} + M\boldsymbol{\Sigma}_A + \sigma^2\boldsymbol{C}_B^{-1}\right)^{-1}. \tag{10}$$

In our previous work, we proved that finding the solution with diagonal covariances is sufficient—any solution has an *equivalent transform* to the solution such that $\boldsymbol{\Sigma}_A$ and $\boldsymbol{\Sigma}_B$ are diagonal (Theorem 1 in Nakajima et al. (2013b)). Under the focus on diagonal covariances, the stationary condition (9) and (10) implies that $\widehat{\boldsymbol{A}}^\top\widehat{\boldsymbol{A}}$ and $\widehat{\boldsymbol{B}}^\top\widehat{\boldsymbol{B}}$ are also diagonal, meaning that the column vectors of $\widehat{\boldsymbol{A}}$, as well as $\widehat{\boldsymbol{B}}$, are orthogonal to each other. Then, we find that the column vectors of $\widehat{\boldsymbol{A}}$ and $\widehat{\boldsymbol{B}}$ only depend on the second term in Eq.(8), which coincides with the objective for (truncated) SVD. Consequently, the mean parameters are expressed as $\widehat{\boldsymbol{a}}_h = \widehat{a}_h\boldsymbol{\omega}_{a_h}$ and $\widehat{\boldsymbol{b}}_h = \widehat{b}_h\boldsymbol{\omega}_{b_h}$ (Lemma 8 in Nakajima and Sugiyama (2011)), and the following proposition thus holds:

Proposition 1 *(Nakajima et al., 2013b) The VB posterior can be written as*

$$r(\boldsymbol{A}, \boldsymbol{B}) = \prod_{h=1}^{H} \mathcal{N}_M(\boldsymbol{a}_h; \widehat{a}_h \boldsymbol{\omega}_{a_h}, \sigma_{a_h}^2 \boldsymbol{I}_M) \mathcal{N}_L(\boldsymbol{b}_h; \widehat{b}_h \boldsymbol{\omega}_{b_h}, \sigma_{b_h}^2 \boldsymbol{I}_L), \tag{11}$$

where $\{\widehat{a}_h, \widehat{b}_h, \sigma_{a_h}^2, \sigma_{b_h}^2\}_{h=1}^H$ *are the solution of the following minimization problem:*

$$\begin{aligned} &\textit{Given} \quad \sigma^2 \in \mathbb{R}_{++}, \quad \{c_{a_h}^2, c_{b_h}^2 \in \mathbb{R}_{++}\}_{h=1}^H, \\ &\min_{\{\widehat{a}_h, \widehat{b}_h, \sigma_{a_h}^2, \sigma_{b_h}^2\}_{h=1}^H} \quad 2F, \\ &\textit{s.t.} \quad \{\widehat{a}_h, \widehat{b}_h \in \mathbb{R}, \quad \sigma_{a_h}^2, \sigma_{b_h}^2 \in \mathbb{R}_{++}\}_{h=1}^H. \end{aligned} \tag{12}$$

Here, F is the free energy (6)*, which can be written as*

$$2F = LM \log(2\pi\sigma^2) + \frac{\sum_{h=1}^L \gamma_h^2}{\sigma^2} + \sum_{h=1}^H 2F_h, \tag{13}$$

$$\begin{aligned} \textit{where} \quad 2F_h = M \log \frac{c_{a_h}^2}{\sigma_{a_h}^2} + L \log \frac{c_{b_h}^2}{\sigma_{b_h}^2} + \frac{\widehat{a}_h^2 + M\sigma_{a_h}^2}{c_{a_h}^2} + \frac{\widehat{b}_h^2 + L\sigma_{b_h}^2}{c_{b_h}^2} \\ - (L+M) + \frac{-2\widehat{a}_h \widehat{b}_h \gamma_h + \left(\widehat{a}_h^2 + M\sigma_{a_h}^2\right)\left(\widehat{b}_h^2 + L\sigma_{b_h}^2\right)}{\sigma^2}. \end{aligned} \tag{14}$$

The minimization problem (12) has been analytically solved (Nakajima et al., 2013b), which provides an analytic-form of the global VB solution (see Proposition 18 in Appendix A). However, the form involves a solution of a *quartic* equation, with which further analysis is difficult. In this paper, finding a shortcut to an alternative *quadratic* equation, we obtain the following theorem, which provides a new and simple analytic-form of the global VB solution (the proof is given in Appendix A):

Theorem 2 *The VB solution can be written as truncated shrinkage SVD as follows:*

$$\widehat{\boldsymbol{U}}^{\mathrm{VB}} = \sum_{h=1}^{H} \widehat{\gamma}_h^{\mathrm{VB}} \boldsymbol{\omega}_{b_h} \boldsymbol{\omega}_{a_h}^\top, \qquad \textit{where} \qquad \widehat{\gamma}_h^{\mathrm{VB}} = \begin{cases} \breve{\gamma}_h^{\mathrm{VB}} & \textit{if } \gamma_h \geq \underline{\gamma}_h^{\mathrm{VB}}, \\ 0 & \textit{otherwise.} \end{cases} \tag{15}$$

Here, the truncation threshold and the shrinkage estimator are, respectively, given by

$$\underline{\gamma}_h^{\mathrm{VB}} = \sigma \sqrt{\frac{(L+M)}{2} + \frac{\sigma^2}{2c_{a_h}^2 c_{b_h}^2} + \sqrt{\left(\frac{(L+M)}{2} + \frac{\sigma^2}{2c_{a_h}^2 c_{b_h}^2}\right)^2 - LM}}, \tag{16}$$

$$\breve{\gamma}_h^{\mathrm{VB}} = \gamma_h \left(1 - \frac{\sigma^2}{2\gamma_h^2}\left(M + L + \sqrt{(M-L)^2 + \frac{4\gamma_h^2}{c_{a_h}^2 c_{b_h}^2}}\right)\right). \tag{17}$$

Our new form with the truncation threshold (16) and the shrinkage estimator (17) consisting of simple algebra facilitates further analysis.

The VB posterior is also written in a simple form (the proof is given in Appendix A):

Corollary 3 *The VB posterior is given by Eq.*(11) *with the following estimators: If* $\gamma_h > \underline{\gamma}_h^{\mathrm{VB}}$,

$$\widehat{a}_h = \pm\sqrt{\breve{\gamma}_h^{\mathrm{VB}}\widehat{\delta}_h^{\mathrm{VB}}}, \qquad \widehat{b}_h = \pm\sqrt{\frac{\breve{\gamma}_h^{\mathrm{VB}}}{\widehat{\delta}_h^{\mathrm{VB}}}}, \qquad \sigma_{a_h}^2 = \frac{\sigma^2\widehat{\delta}_h^{\mathrm{VB}}}{\gamma_h}, \qquad \sigma_{b_h}^2 = \frac{\sigma^2}{\gamma_h\widehat{\delta}_h^{\mathrm{VB}}}, \tag{18}$$

$$\textit{where} \qquad \widehat{\delta}_h^{\mathrm{VB}}\left(\equiv\frac{\widehat{a}_h}{\widehat{b}_h}\right) = \frac{c_{a_h}^2}{\sigma^2}\left(\gamma_h - \breve{\gamma}_h^{\mathrm{VB}} - \frac{L\sigma^2}{\gamma_h}\right), \tag{19}$$

and otherwise,

$$\widehat{a}_h = 0, \qquad \widehat{b}_h = 0, \qquad \sigma_{a_h}^2 = c_{a_h}^2\left(1 - \frac{L\widehat{\zeta}_h^{\mathrm{VB}}}{\sigma^2}\right), \qquad \sigma_{b_h}^2 = c_{b_h}^2\left(1 - \frac{M\widehat{\zeta}_h^{\mathrm{VB}}}{\sigma^2}\right), \tag{20}$$

$$\textit{where} \quad \widehat{\zeta}_h^{\mathrm{VB}}\left(\equiv\sigma_{a_h}^2\sigma_{b_h}^2\right) = \frac{\sigma^2}{2LM}\left(L+M+\frac{\sigma^2}{c_{a_h}^2c_{b_h}^2} - \sqrt{\left(L+M+\frac{\sigma^2}{c_{a_h}^2c_{b_h}^2}\right)^2 - 4LM}\right). \tag{21}$$

3.2 EVB Solution

The empirical VB (EVB) learning, where the hyperparameters $\boldsymbol{C}_A$ and $\boldsymbol{C}_B$ are also estimated from observation, solves the following problem:

$$\begin{aligned} &\text{Given} \quad \sigma^2 \in \mathbb{R}_{++}, \\ &\min_{\{\widehat{a}_h,\widehat{b}_h,\sigma_{a_h}^2,\sigma_{b_h}^2,c_{a_h}^2,c_{b_h}^2\}_{h=1}^H} \quad 2F, \\ &\text{s.t.} \quad \{\widehat{a}_h,\widehat{b}_h \in \mathbb{R}, \quad \sigma_{a_h}^2,\sigma_{b_h}^2,c_{a_h}^2,c_{b_h}^2 \in \mathbb{R}_{++}\}_{h=1}^H. \end{aligned}$$

This problem has also been analytically solved (Nakajima et al., 2013b), which enables efficient computation of the global EVB solution (see Proposition 23 in Appendix B). However, the form requires to solve a quartic equation, and also to evaluate the free energy (14) to judge whether EVB discards each component. This again obstructs further analysis.

By substituting the VB solution, given by Theorem 2 and Corollary 3, we can derive an explicit form of the free energy (13) as a function of $\{c_{a_h}^2, c_{b_h}^2\}_{h=1}^H$ and σ^2. Minimizing it with respect to $\{c_{a_h}^2, c_{b_h}^2\}_{h=1}^H$, we obtain the following theorem, which provides a new and simple analytic-form of the global EVB solution (the proof is given in Appendix B):

Theorem 4 *Let*

$$\alpha = \frac{L}{M} \qquad (0 < \alpha \leq 1), \tag{22}$$

and let $\underline{\tau} = \underline{\tau}(\alpha)$ *be the unique zero-cross point of the following decreasing function:*

$$\Xi(\tau;\alpha) = \Phi(\tau) + \Phi\left(\frac{\tau}{\alpha}\right), \qquad \textit{where} \qquad \Phi(z) = \frac{\log(z+1)}{z} - \frac{1}{2}. \tag{23}$$

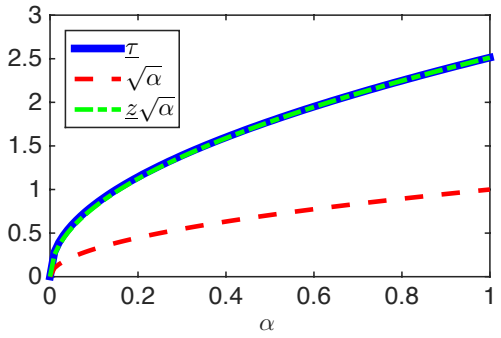


Figure 2: Values of $\underline{\tau}(\alpha)$, $\sqrt{\alpha}$, and $\underline{z}\sqrt{\alpha}$.

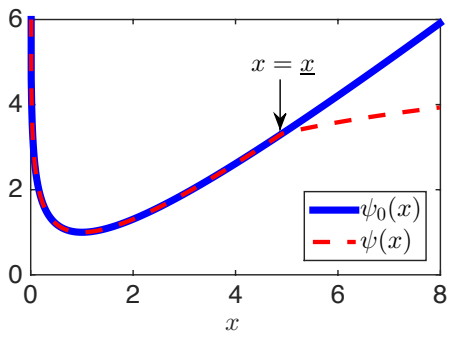


Figure 3: $\psi_0(x)$ and $\psi(x)$.

Then, the EVB solution can be written as truncated shrinkage SVD as follows:

$$\widehat{\boldsymbol{U}}^{\mathrm{EVB}} = \sum_{h=1}^{H} \widehat{\gamma}_h^{\mathrm{EVB}} \boldsymbol{\omega}_{b_h} \boldsymbol{\omega}_{a_h}^{\top}, \qquad \textit{where} \qquad \widehat{\gamma}_h^{\mathrm{EVB}} = \begin{cases} \breve{\gamma}_h^{\mathrm{EVB}} & \textit{if } \gamma_h \geq \underline{\gamma}^{\mathrm{EVB}}, \\ 0 & \textit{otherwise.} \end{cases} \tag{24}$$

Here, the truncation threshold and the shrinkage estimator are, respectively, given by

$$\underline{\gamma}^{\mathrm{EVB}} = \sigma \sqrt{M\,(1+\underline{\tau})\left(1+\frac{\alpha}{\underline{\tau}}\right)}, \tag{25}$$

$$\breve{\gamma}_h^{\mathrm{EVB}} = \frac{\gamma_h}{2}\left(1 - \frac{(M+L)\sigma^2}{\gamma_h^2} + \sqrt{\left(1 - \frac{(M+L)\sigma^2}{\gamma_h^2}\right)^2 - \frac{4LM\sigma^4}{\gamma_h^4}}\right). \tag{26}$$

The EVB threshold (25) involves $\underline{\tau}$, which needs to be numerically computed. However, we can easily prepare a table of the values for $0 < \alpha \leq 1$ beforehand, like the cumulative Gaussian probability used in statistical tests. Alternatively, $\underline{\tau} \approx \underline{z}\sqrt{\alpha}$ is a good approximation, where $\underline{z} \approx 2.5129$ is the unique zero-cross point of $\Phi(z)$, as seen in Figure 2. We can show that $\underline{\tau}$ lies in the following range (see Appendix B for its proof):

$$\sqrt{\alpha} < \underline{\tau} \leq \underline{z}. \tag{27}$$

We will see in Section 5 that $\underline{\tau}$ is an important quantity in describing the behavior of the EVB solution.

In the rest of this section, we summarize some intermediate results obtained in the proof of Theorem 4, which are useful in the subsequent analysis (see Appendix B for their proof):

Corollary 5 *The EVB shrinkage estimator* (26) *is a stationary point of the free energy* (14)*, which exists if and only if*

$$\gamma_h \geq \underline{\gamma}^{\mathrm{local-EVB}} \equiv (\sqrt{L} + \sqrt{M})\sigma, \tag{28}$$

and satisfies the following equation:

$$\left(\gamma_h \breve{\gamma}_h^{\mathrm{EVB}} + L\sigma^2\right)\left(1 + \frac{M\sigma^2}{\gamma_h \breve{\gamma}_h^{\mathrm{EVB}}}\right) = \gamma_h^2. \tag{29}$$

It holds that

$$\gamma_h \breve{\gamma}_h^{\mathrm{EVB}} \geq \sqrt{LM}\sigma^2. \tag{30}$$

Corollary 6 *The minimum free energy achieved under EVB is given by Eq.*(13) *with*

$$2F_h = \begin{cases} M\log\left(\frac{\gamma_h \breve{\gamma}_h^{\mathrm{EVB}}}{M\sigma^2} + 1\right) + L\log\left(\frac{\gamma_h \breve{\gamma}_h^{\mathrm{EVB}}}{L\sigma^2} + 1\right) - \frac{\gamma_h \breve{\gamma}_h^{\mathrm{EVB}}}{\sigma^2} & \text{if } \gamma_h \geq \underline{\gamma}^{\mathrm{EVB}}, \\ 0 & \text{otherwise.} \end{cases} \tag{31}$$

Corollary 5 together with Theorem 4 implies that, when

$$\underline{\gamma}^{\mathrm{local-EVB}} \leq \gamma_h < \underline{\gamma}^{\mathrm{EVB}},$$

a stationary point exists at Eq.(26), but it is not the global minimum. Actually, a local minimum (called a *null* stationary point in Appendix B) with $F_h = 0$ always exists, and the stationary point (26) (called a *positive* stationary point) is a *non-global* local minimum when $\underline{\gamma}^{\mathrm{local-EVB}} < \gamma_h < \underline{\gamma}^{\mathrm{EVB}}$ and the global minimum when $\gamma_h \geq \underline{\gamma}^{\mathrm{EVB}}$ (see Figure 8 and its caption for details). This phase transition induces the free energy thresholding observed in Corollary 6.

We define a *local*-EVB solution by

$$\widehat{\boldsymbol{U}}^{\mathrm{local-EVB}} = \sum_{h=1}^{H} \widehat{\gamma}_h^{\mathrm{local-EVB}} \boldsymbol{\omega}_{b_h} \boldsymbol{\omega}_{a_h}^{\top}, \quad \text{where} \quad \widehat{\gamma}_h^{\mathrm{local-EVB}} = \begin{cases} \breve{\gamma}_h^{\mathrm{EVB}} & \text{if } \gamma_h \geq \underline{\gamma}^{\mathrm{local-EVB}}, \\ 0 & \text{otherwise,} \end{cases} \tag{32}$$

and call $\underline{\gamma}^{\mathrm{local-EVB}}$ a local-EVB threshold. We will discuss an interesting relation between the *local*-EVB solution and an alternative dimensionality selection method (Hoyle, 2008) in Section 6.2.

Rescaling the quantities related to the squared singular value by $M\sigma^2$— to which the contribution from noise (each eigenvalue of $\boldsymbol{\mathcal{E}}^{\top}\boldsymbol{\mathcal{E}}$) scales linearly—simplifies expressions. Assume that the condition (28) holds, and define

$$x_h = \frac{\gamma_h^2}{M\sigma^2}, \tag{33}$$

$$\tau_h = \frac{\gamma_h \breve{\gamma}_h^{\mathrm{EVB}}}{M\sigma^2}, \tag{34}$$

which are used as a rescaled observation and a rescaled EVB estimator, respectively. Eqs.(29) and (26) specify the mutual relations between them:

$$x_h \equiv x(\tau_h; \alpha) = (1 + \tau_h)\left(1 + \frac{\alpha}{\tau_h}\right), \tag{35}$$

$$\tau_h \equiv \tau(x_h;\alpha) = \frac{1}{2}\left(x_h - (1+\alpha) + \sqrt{(x_h-(1+\alpha))^2 - 4\alpha}\right). \tag{36}$$

With these rescaled variables, the condition (28), as well as (30), for the existence of the *positive* local-EVB solution $\breve{\gamma}_h^{\mathrm{EVB}}$ is expressed as

$$x_h \geq \underline{x}^{\mathrm{local}} = \frac{(\underline{\gamma}^{\mathrm{local-EVB}})^2}{M\sigma^2} = x(\sqrt{\alpha};\alpha) = (1+\sqrt{\alpha})^2, \tag{37}$$

$$\tau_h \geq \underline{\tau}^{\mathrm{local}} = \sqrt{\alpha}. \tag{38}$$

The EVB threshold (25) is expressed as

$$\underline{x} = \frac{(\underline{\gamma}^{\mathrm{EVB}})^2}{M\sigma^2} = x(\underline{\tau};\alpha) = (1+\underline{\tau})\left(1+\frac{\alpha}{\underline{\tau}}\right), \tag{39}$$

and the free energy (31) is expressed as

$$F_h = M\tau_h \cdot \min\left(0, \Xi\left(\tau_h;\alpha\right)\right),$$

where $\Xi(\tau;\alpha)$ is defined by Eq.(23).

The rescaled expressions above give an intuition of Theorem 4: The EVB solution $\widehat{\gamma}_h^{\mathrm{EVB}}$ is positive, if and only if the positive local-EVB solution $\breve{\gamma}_h^{\mathrm{EVB}}$ exists (i.e., $x_h \geq \underline{x}^{\mathrm{local}}$), and the free energy $\Xi\left(\tau(x_h;\alpha);\alpha\right)$ at the local-EVB solution is non-positive (i.e., $\tau(x_h;\alpha) \geq \underline{\tau}$ or equivalently $x_h \geq \underline{x}$).

4. Objective Function for Noise Variance Estimation

In this section, we analyze EVB with noise variance estimation:

$$\min_{\{\widehat{a}_h,\widehat{b}_h,\sigma^2_{a_h},\sigma^2_{b_h},c^2_{a_h},c^2_{b_h}\}_{h=1}^H,\sigma^2} \quad 2F,$$
$$\text{s.t.} \quad \{\widehat{a}_h,\widehat{b}_h \in \mathbb{R}, \quad \sigma^2_{a_h},\sigma^2_{b_h},c^2_{a_h},c^2_{b_h} \in \mathbb{R}_{++}\}_{h=1}^H, \sigma^2 \in \mathbb{R}_{++}.$$

Again, by substituting the EVB solution, given by Theorem 4, with the help of Corollary 6, we can express the free energy (13) as a function of the noise variance σ^2. With the rescaled expressions (33)–(39), the free energy is written in a simple form (the proof is given in Appendix C):

Theorem 7 *The noise variance estimator $\widehat{\sigma}^{2\,\mathrm{EVB}}$ is the global minimizer of*

$$\Omega(\sigma^{-2})\left(\equiv \frac{2F(\sigma^{-2})}{LM} + \text{const.}\right) = \frac{1}{L}\left(\sum_{h=1}^{H}\psi\left(\frac{\gamma_h^2}{M\sigma^2}\right) + \sum_{h=H+1}^{L}\psi_0\left(\frac{\gamma_h^2}{M\sigma^2}\right)\right), \tag{40}$$

$$\textit{where} \qquad \psi\left(x\right) = \psi_0\left(x\right) + \theta\left(x > \underline{x}\right)\psi_1\left(x\right), \tag{41}$$

$$\psi_0\left(x\right) = x - \log x, \tag{42}$$

$$\psi_1\left(x\right) = \log\left(\tau(x;\alpha)+1\right) + \alpha\log\left(\frac{\tau(x;\alpha)}{\alpha}+1\right) - \tau(x;\alpha), \tag{43}$$

and $\theta(\cdot)$ denotes an indicator function such that $\theta(\mathit{condition}) = 1$ if the condition is true and $\theta(\mathit{condition}) = 0$ otherwise.

The functions $\psi_0(x)$ and $\psi(x)$ are depicted in Figure 3. We can confirm the convexity of $\psi_0(x)$ and the quasi-convexity of $\psi(x)$,[1] which are useful properties in our analysis.

Let $\widehat{H}^{\mathrm{EVB}}$ be the estimated rank by EVB, i.e., the rank of the EVB estimator $\widehat{U}^{\mathrm{EVB}}$, such that $\widehat{\gamma}_h^{\mathrm{EVB}} > 0$ for $h = 1, \ldots, \widehat{H}^{\mathrm{EVB}}$, and $\widehat{\gamma}_h^{\mathrm{EVB}} = 0$ for $h = \widehat{H}^{\mathrm{EVB}} + 1, \ldots, H$. By bounding the minimizer of the objective (40), we obtain the following theorem (the proof is given in Appendix D):

Theorem 8 *$\widehat{H}^{\mathrm{EVB}}$ is upper-bounded as*

$$\widehat{H}^{\mathrm{EVB}} \leq \overline{H} = \min\left(\left\lceil \frac{L}{1+\alpha} \right\rceil - 1, H\right),$$

and the noise variance estimator $\widehat{\sigma}^{2\,\mathrm{EVB}}$ is bounded as follows:

$$\max\left(\underline{\sigma}^2_{\overline{H}+1}, \frac{\sum_{h=\overline{H}+1}^{L} \gamma_h^2}{M\left(L-\overline{H}\right)}\right) \leq \widehat{\sigma}^{2\,\mathrm{EVB}} \leq \frac{1}{LM}\sum_{h=1}^{L}\gamma_h^2, \tag{44}$$

$$\textit{where} \quad \underline{\sigma}^2_h = \begin{cases} \infty & \textit{for } h = 0, \\ \frac{\gamma_h^2}{M\underline{x}} & \textit{for } h = 1, \ldots, L, \\ 0 & \textit{for } h = L+1. \end{cases} \tag{45}$$

Theorem 8 states that EVB discards the $(L - \lceil L/(1+\alpha)\rceil + 1)$ smallest components, regardless of the observed singular values $\{\gamma_h\}_{h=1}^L$. For example, half of the components are always discarded when the matrix is square (i.e., $\alpha = L/M = 1$). The smallest singular value γ_L is always discarded, and $\widehat{\sigma}^{2\,\mathrm{EVB}} \geq \gamma_L^2/M$ always holds.

Given the EVB estimators $\{\widehat{\gamma}_h^{\mathrm{EVB}}\}_{h=1}^H$ for the singular values, the noise variance estimator $\widehat{\sigma}^{2\,\mathrm{EVB}}$ is specified by the following corollary (the proof is also given in Appendix D):

Corollary 9 *The EVB estimator for the noise variance satisfies the following equality:*

$$\widehat{\sigma}^{2\,\mathrm{EVB}} = \frac{1}{LM}\left(\sum_{l=1}^{L}\gamma_l^2 - \sum_{h=1}^{H}\gamma_h\widehat{\gamma}_h^{\mathrm{EVB}}\right). \tag{46}$$

Theorem 8 and Corollary 9 are used for simple implementation of EVB-PCA in Section 6.1.

5. Performance Analysis

In this section, based on the results obtained in Section 3 and Section 4, we analyze the behavior of EVB with noise variance estimation. We also rely on random matrix theory (Marčenko and Pastur, 1967; Wachter, 1978; Johnstone, 2001; Hoyle and Rattray, 2004; Baik and Silverstein, 2006), which describes the distribution of the singular values of random matrices in the limit when the matrix size goes to infinity. We first introduce some results obtained in random matrix theory, and then apply them to our analysis.

[1] A function $\psi : \mathcal{D} \to \mathbb{R}$ is called quasi-convex if $\psi(\lambda x + (1-\lambda)y) \leq \max(\psi(x), \psi(y)), \forall x, y \in \mathcal{D}, \forall \lambda \in [0,1]$. In other words, $\psi(x)$ is quasi-convex if $-\psi(x)$ is unimodal.

5.1 Random Matrix Theory

Random matrix theory originates from nuclear physics (Wigner, 1957; Mehta, 2000), where the eigenvalue distribution of (infinitely large) symmetric random matrices was investigated to analyze the spectra of heavy atoms. In statistical applications, Wishart matrices play an important role, of which the eigenvalue distribution (or equivalently, the singular value distribution of random data matrices) was derived (Marčenko and Pastur, 1967; Wachter, 1978). Under appropriate scaling, those distributions typically have a finite support, which enables us to clean noisy data and bound quantities related to randomness. Results from random matrix theory have been used in many research fields, including financial risk analysis, where the observed covariance matrix is cleaned for stable prediction (Bouchaud and Potters, 2003), information theory, where the capacity of noisy communication channel was evaluated (Tulino and Verdu, 2004), and signal processing, where the restricted isometry property of random projection was proved for guaranteeing the performance of compressed sensing (Candès and Tao, 2006; Recht et al., 2010). Development of random matrix theory is still actively on going, and new important results are being reported (Bai and Silverstein, 2010).

To analyze the performance of EVB-PCA, we assume that the observed matrix $\boldsymbol{V}$ is generated from the *spiked covariance* model (Johnstone, 2001):

$$\boldsymbol{V} = \boldsymbol{U}^* + \boldsymbol{\mathcal{E}},$$

where $\boldsymbol{U}^* \in \mathbb{R}^{L \times M}$ is a *true* signal matrix with rank H^* and singular values $\{\gamma_h^*\}_{h=1}^{H^*}$, and $\boldsymbol{\mathcal{E}} \in \mathbb{R}^{L \times M}$ is a random matrix such that each element is independently drawn from a distribution with mean zero and variance σ^{*2} (not necessarily Gaussian). As the observed singular values $\{\gamma_h\}_{h=1}^{L}$ of $\boldsymbol{V}$, the true singular values $\{\gamma_h^*\}_{h=1}^{H^*}$ are also assumed to be arranged in the non-increasing order.

We define rescaled versions of the observed and the true singular values:

$$\begin{aligned} y_h &= \frac{\gamma_h^2}{M\sigma^{*2}} && \text{for} && h = 1, \ldots, L, \\ \nu_h^* &= \frac{\gamma_h^{*2}}{M\sigma^{*2}} && \text{for} && h = 1, \ldots, H^*. \end{aligned}$$

In other words, $\{y_h\}_{h=1}^{L}$ are the eigenvalues of $\boldsymbol{V}\boldsymbol{V}^\top/(M\sigma^{*2})$, and $\{\nu_h^*\}_{h=1}^{H^*}$ are the eigenvalues of $\boldsymbol{U}^*\boldsymbol{U}^{*\top}/(M\sigma^{*2})$. Note the difference between x_h, defined by Eq.(33), and y_h: x_h is the squared observed singular value rescaled with the model noise variance σ^2 to be estimated, while y_h is the one rescaled with the true noise variance σ^{*2}.

Define the empirical distribution of the observed eigenvalues $\{y_h\}_{h=1}^{L}$ by

$$p(y) = \frac{1}{L}\sum_{h=1}^{L} \delta(y - y_h),$$

where $\delta(y)$ denotes the Dirac delta function. When $H^* = 0$, the observed matrix $\boldsymbol{V} = \boldsymbol{\mathcal{E}}$ consists only of noise, and its singular value distribution in the large-scale limit is specified by the following proposition:

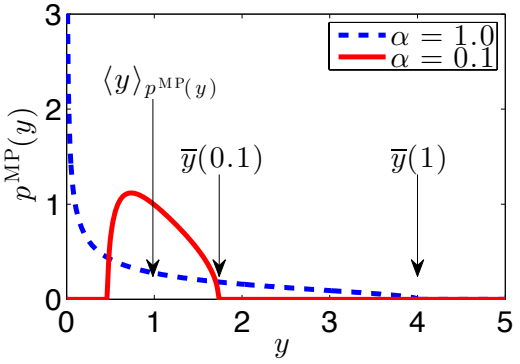


Figure 4: Marčenko-Pastur distribution.

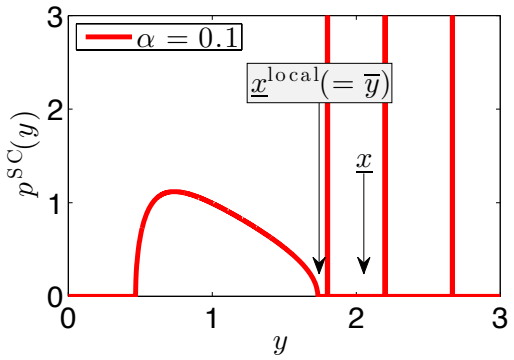


Figure 5: Spiked covariance distribution when $\{\nu_h^*\}_{h=1}^{H^{**}} = \{1.5, 1.0, 0.5\}$.

Proposition 10 *(Marčenko and Pastur, 1967; Wachter, 1978) In the large-scale limit when L and M go to infinity with its ratio $\alpha = L/M$ fixed, the empirical distribution of the eigenvalue y of $\boldsymbol{\mathcal{E}}\boldsymbol{\mathcal{E}}^\top/(M\sigma^{*2})$ converges almost surely to*

$$p(y) \to p^{\mathrm{MP}}(y) \equiv \frac{\sqrt{(y-\underline{y})(\overline{y}-y)}}{2\pi\alpha y}\theta(\underline{y} < y < \overline{y}), \tag{47}$$

$$\textit{where} \qquad \overline{y} = (1+\sqrt{\alpha})^2, \qquad \underline{y} = (1-\sqrt{\alpha})^2, \tag{48}$$

and $\theta(\cdot)$ is the indicator function, defined in Theorem 7.

Figure 4 shows Eq.(47), which we call the Marčenko-Pastur (MP) distribution, for $\alpha = 0.1, 1$. The mean $\langle y \rangle_{p^{\mathrm{MP}}(y)} = 1$ (which is constant for any $0 < \alpha \leq 1$) and the upper-limits $\overline{y} = \overline{y}(\alpha)$ of the support for $\alpha = 0.1, 1$ are indicated by arrows. Proposition 10 states that the probability mass is concentrated in the range between $\underline{y} \leq y \leq \overline{y}$. Note that the MP distribution appears for a *single* sample matrix; different from standard "large-sample" theories, Proposition 10 does not require to average over many sample matrices (this property is called *self-averaging*). This single-sample property of the MP distribution is highly useful in our analysis because we are working with a single observation matrix in the PCA scenario.

When $H^* > 0$, the true signal matrix $\boldsymbol{U}^*$ affects the singular value distribution of $\boldsymbol{V}$. However, if $H^* \ll L$, the distribution can be approximated by a mixture of spikes (delta functions) and the MP distribution $p^{\mathrm{MP}}(y)$. Let H^{**} $(\leq H^*)$ be the number of singular values of $\boldsymbol{U}^*$ greater than $\gamma_h^* > \alpha^{1/4}\sqrt{M}\sigma^*$, i.e.,

$$\nu_{H^{**}}^* > \sqrt{\alpha} \qquad \text{and} \qquad \nu_{H^{**}+1}^* \leq \sqrt{\alpha}.$$

Then, the following proposition holds:

Proposition 11 *(Baik and Silverstein, 2006) In the large-scale limit when L and M go to infinity with finite α and H^*, it almost surely holds that*

$$y_h = y_h^{\mathrm{Sig}} \equiv (1+\nu_h^*)\left(1+\frac{\alpha}{\nu_h^*}\right) \quad \text{for} \quad h = 1, \ldots, H^{**}, \tag{49}$$

$$y_{H^{**}+1} = \overline{y}, \quad \text{and} \quad y_L = \underline{y}.$$

Furthermore, Hoyle and Rattray (2004) argued that, when L and M are large (but finite) and $H^* \ll L$, the empirical distribution of the eigenvalue y of $\boldsymbol{V}\boldsymbol{V}^\top/(M\sigma^{*2})$ is accurately approximated by

$$p(y) \approx p^{\mathrm{SC}}(y) \equiv \frac{1}{L}\sum_{h=1}^{H^{**}} \delta\left(y - y_h^{\mathrm{Sig}}\right) + \frac{L-H^{**}}{L} p^{\mathrm{MP}}(y). \tag{50}$$

Figure 5 shows Eq.(50), which we call the spiked covariance (SC) distribution, for $\alpha = 0.1$, $H^{**} = 3$, and $\{\nu_h^*\}_{h=1}^{H^{**}} = \{1.5, 1.0, 0.5\}$. The SC distribution is irrespective of $\{\nu_h^*\}_{h=H^{**}+1}^{H^*}$, which satisfy $0 < \nu_h^* \leq \sqrt{\alpha}$ by definition.

Proposition 11 states that, in the large-scale limit, the large signal components such that $\nu_h^* > \sqrt{\alpha}$ appear outside the support of the MP distribution as spikes, while the other small signals are indistinguishable from the MP distribution (note that $y_h^{\mathrm{Sig}} > \overline{y}$ for $\nu_h^* > \sqrt{\alpha}$). This implies that any PCA method fails to recover the true dimensionality, unless

$$\nu_{H^*}^* > \sqrt{\alpha}. \tag{51}$$

The condition (51) requires that U^* has no small positive singular value such that $0 < \nu_h^* \leq \sqrt{\alpha}$, and therefore $H^{**} = H^*$.

The approximation (50) allows us to investigate more practical situations when the matrix size is finite. Based on this approximation, Hoyle (2008) analyzed the performance of the *overlap* method, an alternative dimensionality selection method which will be introduced and discussed in Section 6.2. In Section 5.2, we provide two theorems: One is based on Proposition 11, and guarantees the perfect dimensionality recovery of EVB in the large-scale limit, and the other one relies on the approximation (50), and provides a more realistic condition for perfect recovery.

5.2 Perfect Dimensionality Recovery Condition

Now, we are almost ready for clarifying the behavior of EVB-PCA. We assume that the model rank is set to be large enough, i.e., $H^* \leq H \leq L$, and all model parameters including the noise variance are estimated from observation. The last proposition on which our analysis relies is related to the property, called the *strong unimodality*,[2] of the log-concave distributions:

Proposition 12 *(Ibragimov, 1956; Dharmadhikari and Joag-dev, 1988) The convolution*

$$g(s) = \langle f(s+t) \rangle_{p(t)} = \int f(s+t)p(t)dt$$

is quasi-convex, if $p(t)$ is a log-concave distribution, and $f(t)$ is a quasi-convex function.

[2] A distribution $p(t)$ is called strongly unimodal if the convolution of $p(t)$ with any unimodal function is unimodal.

In the large-scale limit, the summation over $h = 1, \ldots, L$ in the objective $\Omega(\sigma^{-2})$, given by Eq.(40), for noise variance estimation can be replaced with an expectation over the MP distribution $p^{\mathrm{MP}}(y)$. By scaling variables, the objective can be written as a convolution with a scaled version of the MP distribution, which turns out to be log-concave. Accordingly, we can use Proposition 12 to show that $\Omega(\sigma^{-2})$ is quasi-convex, and therefore, the noise variance estimation by EVB is accurate. Combining this result with Proposition 11, we obtain the following theorem (the proof is given in Appendix E):

Theorem 13 *In the large-scale limit when L and M go to infinity with finite α and H^*, EVB almost surely recovers the true rank, i.e., $\widehat{H}^{\mathrm{EVB}} = H^*$, if and only if*

$$\nu^*_{H^*} \geq \underline{\tau}, \tag{52}$$

where $\underline{\tau}$ is defined in Theorem 4.

Furthermore, the following corollary completely describes the behavior of EVB in the large-scale limit (the proof is also given in Appendix E):

Corollary 14 *In the large-scale limit, the objective $\Omega(\sigma^{-2})$, defined by Eq.(40), for the noise variance estimation converges to a quasi-convex function, and it almost surely holds that*

$$\widehat{\tau}^{\mathrm{EVB}}_h \left(\equiv \frac{\gamma_h \widehat{\gamma}^{\mathrm{EVB}}_h}{M \widehat{\sigma}^{2\,\mathrm{EVB}}} \right) = \begin{cases} \nu^*_h & \text{if } \nu^*_h \geq \underline{\tau}, \\ 0 & \text{otherwise}, \end{cases} \tag{53}$$
$$\widehat{\sigma}^{2\,\mathrm{EVB}} = \sigma^{*2}.$$

One may get intuition of Eqs.(52) and (53) from comparing Eqs.(39) and (35) with Eq.(49): The estimator τ_h has the same relation to the observation x_h as the true signal ν^*_h, and hence is an unbiased estimator of the signal. However, Theorem 13 does not even approximately hold in practical situations with moderate-sized matrices (see the numerical simulation below). The following theorem, which relies on the approximation (50), provides a more practical condition for perfect recovery (the proof is given in Appendix F):

Theorem 15 *Let*

$$\xi = \frac{H^*}{L}$$

be the relevant rank (dimensionality) ratio, and assume that

$$p(y) = p^{\mathrm{SC}}(y). \tag{54}$$

Then, EVB recovers the true rank, i.e., $\widehat{H}^{\mathrm{EVB}} = H^*$, if the following two inequalities hold:

$$\xi < \frac{1}{\underline{x}}, \tag{55}$$

$$\nu^*_{H^*} > \frac{\left(\frac{\underline{x}-1}{1-\underline{x}\xi} - \alpha \right) + \sqrt{\left(\frac{\underline{x}-1}{1-\underline{x}\xi} - \alpha \right)^2 - 4\alpha}}{2}, \tag{56}$$

where $\underline{x}$ is defined by Eq.(39).

Note that, in the large-scale limit, ξ converges to zero, and the sufficient condition, (55) and (56), in Theorem 15 is equivalent to the necessary and sufficient condition (52) in Theorem 13.

Theorem 15 only requires that the SC distribution (50) well approximates the observed singular value distribution. Accordingly, it well describes the dependency of the behavior of EVB on ξ, as shown in the numerical simulation below. Theorem 15 states that, if the true rank H^* is small enough compared with L and the smallest signal $\nu^*_{H^*}$ is large enough, EVB perfectly recovers the true dimensionality.

The following corollary also supports EVB (the proof is also given in Appendix F):

Corollary 16 *Under the assumption* (54) *and the conditions* (55) *and* (56)*, the objective* $\Omega(\sigma^{-2})$ *for the noise variance estimation has no local minimum (no stationary point if* $\xi > 0$*) that results in a wrong estimated rank* $\widehat{H}^{\mathrm{EVB}} \neq H^*$*.*

This corollary states that, although the objective function (40) is non-convex and possibly multimodal in general, any local minimum leads to the correct estimated rank. Therefore, perfect recovery does not require global search, but only local search, for noise variance estimation, if L and M are sufficiently large so that we can assume Eq.(54).

Figure 6 shows numerical simulation results for $M = 200$ and $L = 20, 100, 200$. $\boldsymbol{\mathcal{E}}$ was drawn from the independent Gaussian distribution with variance $\sigma^{*2} = 1$, and *true* signal singular values $\{\gamma^*_h\}_{h=1}^{H^*}$ were drawn from the uniform distribution on $[z\sqrt{M}\sigma^*, 10\sqrt{M}\sigma^*]$ for different z, which is indicated by the horizontal axis. The vertical axis indicates the success rate of dimensionality recovery, i.e., $\widehat{H}^{\mathrm{EVB}} = H^*$, over 100 trials. If the condition (55) on ξ is violated, the corresponding curve is depicted with markers. Otherwise, the condition (56) on $\nu^*_{H^*}(= \gamma^{*2}_{H^*}/(M\sigma^{*2}))$ is indicated by a vertical bar with the same color and line style for each ξ. In other words, Theorem 15 states that the success rate should be equal to one if z $(> \gamma^*_{H^*}/(\sqrt{M}\sigma^*))$ is larger than the value indicated by the vertical bar. The solid cyan bar, which lies at the left-most in each graph, indicates the condition (52) given by Theorem 13.

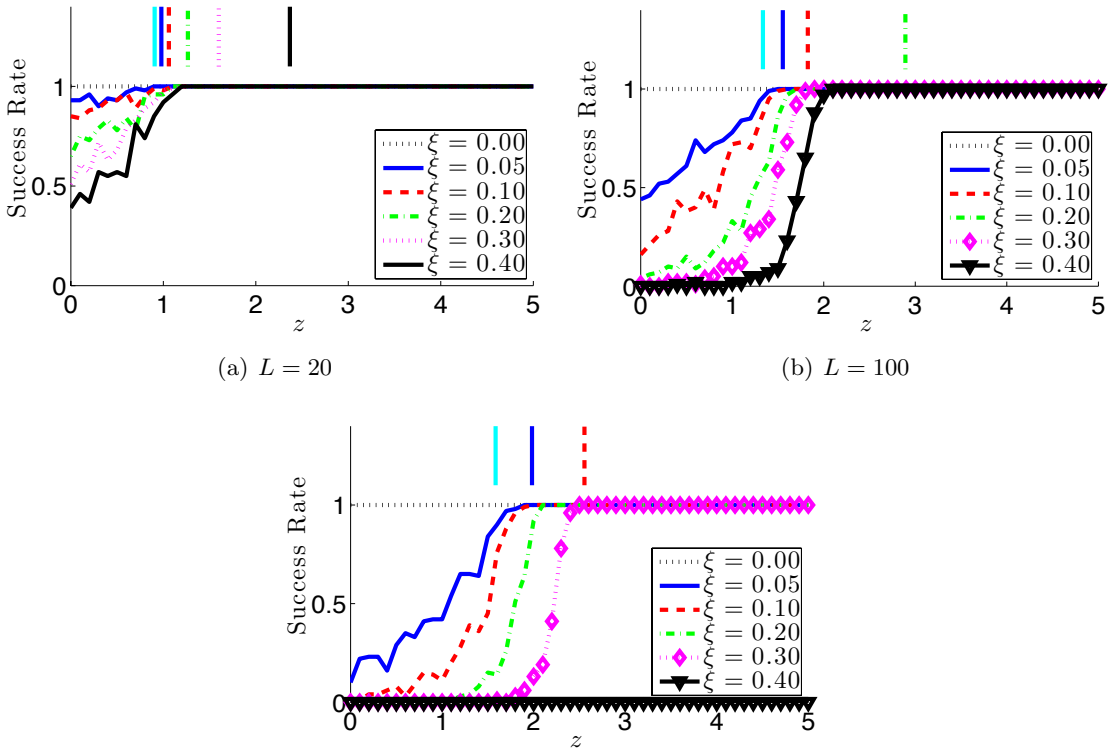
We see that Theorem 15 with the condition (56) approximately holds for these moderate-sized matrices, while Theorem 13 with the condition (52), which does not depend on the relevant rank ratio ξ, immediately breaks for positive ξ.

6. Discussion

In this section, we first propose a few implementations of EVB-PCA. After that, by contrasting with an alternative dimensionality selection method, we characterize the behavior of EVB-PCA, and discuss the optimality in the large-scale limit.

6.1 Implementation

The analytic-form solution derived in Nakajima et al. (2013b) involves a solution of a *quartic* equation. To implement EVB-PCA based on that form, we needed to use a highly complicated analytic-form solution, derived by, e.g., Ferrari's method, or rely on a numerical quartic solver. Our new analytic-form solution can greatly simplify the implementation. Note that, since our theory of performance guarantee assumes that the observed matrix has no missing entry, its applicability is mostly limited to the standard use of PCA—dimensionality



(a) $L = 20$

(b) $L = 100$

(c) $L = 200$

Figure 6: Success rate of dimensionality recovery in numerical simulation for $M = 200$. The horizontal axis indicates the lower limit of the support of the simulated true signal distribution, i.e., $z \approx \sqrt{\nu^*_{H^*}}$. The recovery condition (56) for finite-sized matrices is indicated by a vertical bar with the same color and line style for each ξ. The recovery condition (52), which does not depend on ξ, for infinite-sized matrices is also indicated by a solid cyan bar.

reduction for preprocessing (Bishop, 2006). However, our simple implementation introduced below can be applied to more general cases where the global VB solver is used as a subroutine, e.g., in non-conjugate matrix factorization with missing entries (Seeger and Bouchard, 2012), and in *sparse additive matrix factorization* (Nakajima et al., 2013a), an extension of robust PCA.

A table of $\underline{\tau}$ defined in Theorem 4 should be prepared beforehand (or use a simple approximation $\underline{\tau} \approx \underline{z}\sqrt{\alpha} \approx 2.5129\sqrt{\alpha}$). Given an observed matrix $\boldsymbol{V}$, we perform SVD and obtain the singular values $\{\gamma_h\}_{h=1}^{L}$. After that, in our new implementation, we first directly estimate the noise variance based on Theorem 7, using any 1-D local search algorithm with the search range restricted by Theorem 8. Thus, we obtain the noise variance estimator $\widehat{\sigma}^{2\,\mathrm{EVB}}$. Discarding all the components such that $\underline{\sigma}^2_h < \widehat{\sigma}^{2\,\mathrm{EVB}}$, where $\underline{\sigma}^2_h$ is defined by

Algorithm 1 Global EVB-PCA algorithm.

1: Transpose $\boldsymbol{V} \to \boldsymbol{V}^\top$ if $L > M$.
2: Refer to the table of $\underline{\tau}(\alpha)$ at $\alpha = L/M$ (or use a simple approximation $\underline{\tau} \approx 2.5129\sqrt{\alpha}$).
3: Set H ($\leq L$) to a sufficiently large value, and compute the SVD of $\boldsymbol{V} = \sum_{h=1}^{H} \gamma_h \boldsymbol{\omega}_{b_h} \boldsymbol{\omega}_{a_h}^\top$.
4: Locally search the minimizer $\widehat{\sigma}^{2\,\mathrm{EVB}}$ of Eq.(40), which lies in the range (44).
5: Discard the components such that $\underline{\sigma}_h^2 < \widehat{\sigma}^{2\,\mathrm{EVB}}$, where $\underline{\sigma}_h^2$ is defined by Eq.(45).

Eq.(45), gives a dimensionality reduction result. Algorithm 1 describes a pseudo code.[3] If necessary, Theorem 4 gives the EVB estimator $\widehat{\boldsymbol{U}}^{\mathrm{EVB}}$ for $\sigma^2 = \widehat{\sigma}^{2\,\mathrm{EVB}}$. The EVB posterior is also easily computed by using Corollary 3. In this way, we can easily perform EVB-PCA equipped with the guaranteed automatic dimensionality selection functionality at little expense—computation time of Algorithm 1 is dominated by SVD, which the plain PCA also requires to perform.

Another implementation, which we refer to as EVB(Ite), is to iterate Eqs.(24) and (46) in turn. Although it is not guaranteed, EVB(Ite) tends to converge to the global solution if we initialize the noise variance $\widehat{\sigma}^{2\,\mathrm{EVB}}$ sufficiently small (see Section 6.2).

Finally, we introduce an iterative algorithm for the local-EVB solution, defined by Eq.(32). This solution can be obtained by iterating Eq.(32) and

$$\widehat{\sigma}^{2\,\mathrm{local-EVB}} = \frac{1}{LM}\left(\sum_{l=1}^{L}\gamma_l^2 - \sum_{h=1}^{H}\gamma_h \widehat{\gamma}_h^{\mathrm{local-EVB}}\right) \tag{57}$$

in turn. If we initialize the noise variance $\widehat{\sigma}^{2\,\mathrm{local-EVB}}$ sufficiently small, this algorithm can be trapped at the *positive* stationary point for each h even if it is not the global minimum, and tends to converge to the local-EVB solution.

6.2 Comparison with Laplace Approximation

Here, we compare EVB with the *overlap* method (Hoyle, 2008), an alternative dimensionality selection method based on the Laplace approximation (LA). Consider the PCA application, where D denotes the dimensionality of the observation space, and N denotes the number of samples, i.e., in our MF notation to keep $L \leq M$,

$$\begin{aligned} L = D, M = N \qquad & \text{if} \qquad D \leq N, \\ L = N, M = D \qquad & \text{if} \qquad D > N. \end{aligned}$$

Just after Tipping and Bishop (1999) proposed the probabilistic PCA, Bishop (1999b) proposed to select the PCA dimension by maximizing the marginal likelihood:[4]

$$p(\boldsymbol{V}) = \langle p(\boldsymbol{V}|\boldsymbol{A},\boldsymbol{B})\rangle_{p(\boldsymbol{A})p(\boldsymbol{B})}\,. \tag{58}$$

[3] The MATLAB® code will be available at `http://sites.google.com/site/shinnkj23/`.

[4] Tipping and Bishop (1999) adopted *partially Bayesian* (PB) learning, where $\boldsymbol{A}$ is marginalized out and $\boldsymbol{B}$ is point-estimated. Although PB has some similarities to VB (Nakajima et al., 2011; Nakajima and Sugiyama, 2014), it does not offer automatic dimensionality selection when all hyperparameters $(\boldsymbol{C}_A, \boldsymbol{C}_B, \sigma^2)$ are unknown.

Since the marginal likelihood (58) is computationally intractable, he approximated it by LA, and suggested Gibbs sampling and VB learning as alternatives. The VB variant, of which the model is almost the same as ours (1)–(3), was proposed by himself (Bishop, 1999a). A standard local search algorithm, where the means and the covariances of $\boldsymbol{A}$ and $\boldsymbol{B}$ are iteratively updated, was used for inference.

The LA-based approach was polished in Minka (2001), by introducing a conjugate prior on $\boldsymbol{B}$ to $p(\boldsymbol{V}|\boldsymbol{B}) = \langle p(\boldsymbol{V}|\boldsymbol{A}, \boldsymbol{B})\rangle_{p(\boldsymbol{A})}$, and ignoring the non-leading terms that do not grow fast as the number N of samples goes to infinity. Hoyle (2008) pointed out that Minka's method is inaccurate when $D \gg N$, and proposed the overlap (OL) method, a further polished variant of the LA-based approach. A notable difference of OL from most of the LA-based methods is that OL applies LA to a more accurate estimator than the MAP estimator, while the other methods apply LA simply to the MAP estimator. Thanks to the use of an accurate estimator, OL behaves *optimally* in the large-scale limit when D and N go to infinity, while Minka's method does not. We will clarify the meaning of optimality, and discuss it in more detail in Section 6.3.

OL minimizes an approximation to the negative log of the marginal likelihood (58), which depends on estimators of $\lambda_h = b_h^2 + \sigma^2$ and σ^2 computed by an iterative algorithm, over the hypothetical model rank $H = 1, \ldots, L$ (see Appendix H for details). Figure 7 shows numerical simulation results that compare EVB and OL: Figure 7(a) shows the success rate for the no signal case $\xi = 0$ ($H^* = 0$), while Figures 7(b)–7(f) show the success rate for $\xi = 0.05$ and $D = 20, 100, 200, 400, 1000$, respectively.

We also show the performance of EVB(Ite) and local-EVB. As mentioned in Section 6.1, EVB(Ite) gives almost the same results as EVB. Local-EVB behaves similarly to OL except the case when D/N is small (Figure 7(b)). The reason of this similarity will be elucidated in Section 6.3. For OL, EVB(Ite), and local-EVB, we initialized the noise variance estimator to $10^{-4} \cdot \sum_{h=1}^{L} \gamma_h^2/(LM)$.

Comparing EVB with OL, we observe the conservative nature of EVB: It exhibits almost zero false positive rate at the expense of low sensitivity. Because of the low sensitivity, EVB actually does not behave optimally in the large-scale limit, which is discussed in Section 6.3.

6.3 Optimality in Large-scale Limit

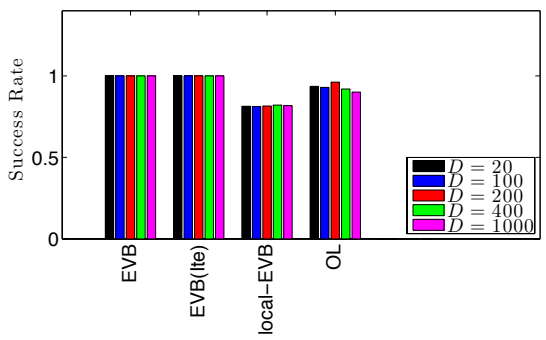
Consider the large-scale limit, i.e., $L, M \to \infty, \alpha = L/M$, and assume that the model rank H is set to be large enough but finite so that $H \geq H^*$ and $H/L \to 0$. Then, OL is equivalent to counting the number of components such that $\widehat{\lambda}_h^{\mathrm{OL-limit}} > \widehat{\sigma}^{2\,\mathrm{OL-limit}}$, i.e.,

$$\widehat{H}^{\mathrm{OL-limit}} = \sum_{h=1}^{L} \theta\left(\widehat{\lambda}_h^{\mathrm{OL-limit}} > \widehat{\sigma}^{2\,\mathrm{OL-limit}}\right), \tag{59}$$

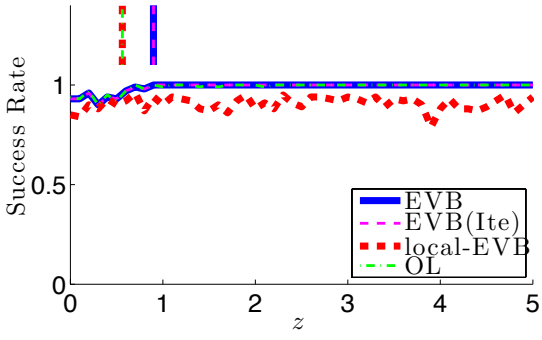
after the following updates converge:

$$\widehat{\lambda}_h^{\mathrm{OL-limit}} = \begin{cases} \breve{\lambda}_h^{\mathrm{OL-limit}} & \text{if } \gamma_h \geq \underline{\gamma}^{\mathrm{local-EVB}}, \\ \widehat{\sigma}^{2\,\mathrm{OL-limit}} & \text{otherwise,} \end{cases} \quad \text{for} \quad h = 1, \ldots, H, \tag{60}$$

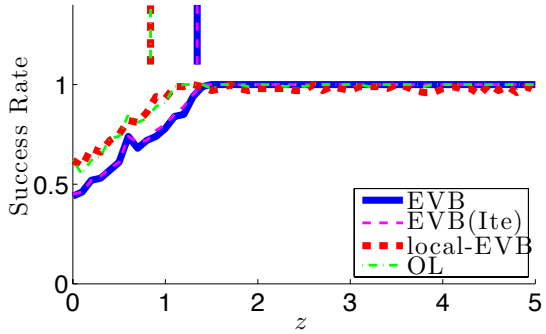
$$\widehat{\sigma}^{2\,\mathrm{OL-limit}} = \frac{1}{(M-H)}\left(\sum_{l=1}^{L}\frac{\gamma_l^2}{L} - \sum_{h=1}^{H}\widehat{\lambda}_h^{\mathrm{OL-limit}}\right), \tag{61}$$



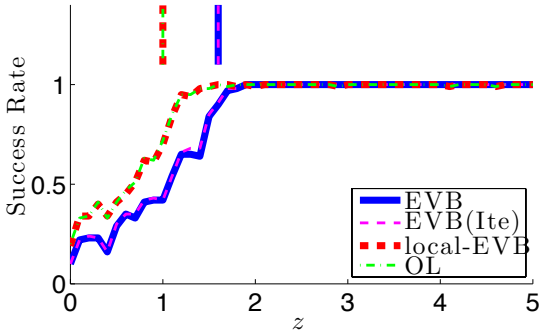
(a) $\xi = 0$



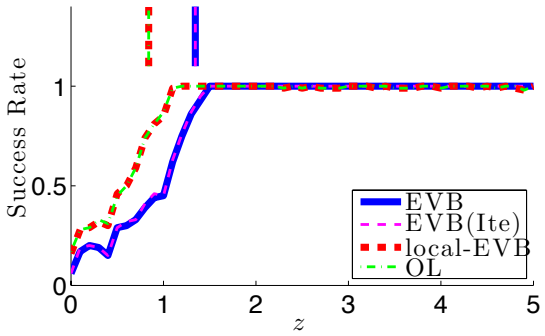
(b) $\xi = 0.05, D = 20$



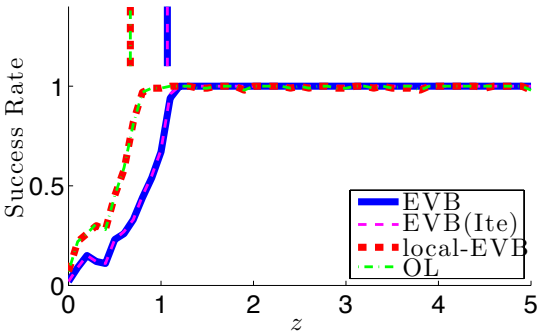
(c) $\xi = 0.05, D = 100$



(d) $\xi = 0.05, D = 200$



(e) $\xi = 0.05, D = 400$



(f) $\xi = 0.05, D = 1000$

Figure 7: Success rate of dimensionality recovery by EVB, EVB(Ite), local-EVB, and OL for $N = 200$. Vertical bars indicate the recovery conditions, Eq.(52) for EVB and EVB(Ite), and Eq.(63) for OL and local-EVB, in the large-scale limit.

where $\breve{\lambda}_h^{\mathrm{OL-limit}} = \dfrac{\gamma_h^2}{2L}\left(1 - \dfrac{(M-L)\widehat{\sigma}^{2\,\mathrm{OL-limit}}}{\gamma_h^2}\right.$

$$+\sqrt{\left(1-\frac{(M-L)\widehat{\sigma}^{2\ \mathrm{OL-limit}}}{\gamma_h^2}\right)^2-\frac{4L\widehat{\sigma}^{2\ \mathrm{OL-limit}}}{\gamma_h^2}}\Bigg). \quad (62)$$

OL evaluates its objective, which approximates the negative log of the marginal likelihood (58), after the updates (60) and (61) converge for each hypothetical H, and adopts the minimizer $\widehat{H}^{\mathrm{OL-limit}}$ as the rank estimator. However, Hoyle (2008) proved that, in the large-scale limit, the objective decreases as H increases, as long as Eq.(62) is a real number (or equivalently $\gamma_h \geq \underline{\gamma}^{\mathrm{local-EVB}}$ holds) for all $h = 1, \ldots, H$ at the convergence. Accordingly, Eq.(59) suffices.

Interestingly, the threshold in Eq.(60) coincides with the local-EVB threshold (28). Moreover, the updates (60) and (61) for OL are equivalent to the updates (32) and (57) for local-EVB with the following correspondence:

$$\widehat{\lambda}_h^{\mathrm{OL-limit}} = \frac{\gamma_h \widehat{\gamma}_h^{\mathrm{local-EVB}}}{L} + \widehat{\sigma}^{2\ \mathrm{local-EVB}},$$
$$\widehat{\sigma}^{2\ \mathrm{OL-limit}} = \widehat{\sigma}^{2\ \mathrm{local-EVB}}.$$

Thus, the dimensionality selection by OL and local-EVB are equivalent in the large-scale limit, i.e., $\widehat{H}^{\mathrm{OL-limit}} = \widehat{H}^{\mathrm{local-EVB}}$.

The optimality of OL in the large-scale limit was shown:

Proposition 17 *(Hoyle, 2008) In the large-scale limit when L and M go to infinity with finite α, H^*, and H ($\geq H^*$)[5], OL almost surely recovers the true rank, i.e., $\widehat{H}^{\mathrm{OL-limit}} = H^*$, if and only if*

$$\nu_{H^*}^* > \sqrt{\alpha}. \quad (63)$$

It almost surely holds that

$$\frac{\widehat{\lambda}_h^{\mathrm{OL-limit}}}{\widehat{\sigma}^{2\ \mathrm{OL-limit}}} - 1 = \nu_h^*,$$
$$\widehat{\sigma}^{2\ \mathrm{OL-limit}} = \sigma^{*2}.$$

Note that the condition (63) coincides with the condition (51)—random matrix theory states that any signal component violating this condition is indistinguishable from the noise distribution, and therefore, any PCA method fails to recover the correct dimensionality if such a signal component exists. In this sense, OL, as well as local-EVB, is *optimal* in the large-scale limit.

On the other hand, Theorem 13 implies that (global) EVB is not optimal in the large-scale limit, and more conservative (see the difference between $\underline{\tau}$ and $\sqrt{\alpha}$ in Figure 2). In Figure 7, the conditions for perfect dimensionality recovery in the large-scale limit are indicated by vertical bars:

$$z = \sqrt{\underline{\tau}} \ \text{ for EVB and EVB(Ite)}, \qquad \text{and} \qquad z = \sqrt{\underline{\tau}^{\mathrm{local}}} = \alpha^{1/4} \ \text{ for OL and local-EVB.}$$

[5] Unlike our analysis in Section 5, Hoyle (2008) assumes that $H/L \to 0$, which trivially guarantees that the noise variance is accurately estimated.

All methods accurately estimate the noise variance in the large-scale limit, i.e.,

$$\widehat{\sigma}^{2\,\mathrm{EVB}} = \widehat{\sigma}^{2\,\mathrm{OL-limit}} = \widehat{\sigma}^{2\,\mathrm{local-EVB}} = \sigma^{*2}.$$

Taking this into account, we indicate the recovery conditions in Figure 5 by arrows at

$$y = \underline{x} \ \text{ for EVB and EVB(Ite)}, \qquad \text{and} \qquad y = \underline{x}^{\mathrm{local}} (= \overline{y}) \ \text{ for OL and local-EVB},$$

respectively. Figure 5 implies that, in this particular case, EVB discards the third spike coming from the third true signal $\nu_3^* = 0.5$, while OL and local-EVB successfully capture it as a signal.

When the matrix size is finite, the conservative nature of EVB is not always bad, since it offers almost zero false positive rate, which makes Theorem 15 approximately hold for finite cases, as seen in Figure 6 and Figure 7. However, the fact that not (global) EVB but local-EVB is optimal in the large-scale limit should be a consequence of inaccurate approximation of VB learning under the independence assumption. We will further investigate the difference between VB and Bayesian learning in our future work.

7. Conclusion

In this paper, we analyzed the variational Bayesian (VB) learning in probabilistic PCA. More specifically, we considered empirical VB (EVB) learning with noise variance estimation, i.e., all model parameters are estimated from observed data. We established a necessary and sufficient condition for perfect dimensionality recovery by EVB-PCA, which theoretically guarantees its performance. At the same time, our result also revealed the conservative nature of EVB-PCA—it offers a low false positive rate at the expense of low sensitivity, due to which EVB-PCA does not behave optimally in the large-scale limit.

By contrasting with an alternative dimensionality selection method, called the overlap (OL) method, we characterized the behavior of EVB. We also pointed out the equivalence between OL and local-EVB, a slight modification of EVB, in the large scale limit.

In our analysis, we derived bounds of the noise variance estimator, and a new and simple analytic-form solution for the other parameters, with which we proposed a new simple implementation of EVB-PCA.

Acknowledgments

The authors thank anonymous reviewers for helpful comments. Shinichi Nakajima thanks the support from Nikon Corporation, the support from Grant-in-Aid for Scientific Research on Innovative Areas: Prediction and Decision Making, 23120004, and the support from the German Research Foundation (GRK 1589/1) by the Federal Ministry of Education and Research (BMBF) under the Berlin Big Data Center project (FKZ 01IS14013A). Masashi Sugiyama was supported by the CREST program.

Appendix A. Proof of Theorem 2 and Corollary 3

The global VB solution is known:

Proposition 18 *(Nakajima et al., 2013b) The VB solution can be written as truncated shrinkage SVD as follows:*

$$\widehat{\boldsymbol{U}}^{\mathrm{VB}} = \sum_{h=1}^{H} \widehat{\gamma}_h^{\mathrm{VB}} \boldsymbol{\omega}_{b_h} \boldsymbol{\omega}_{a_h}^{\top}, \qquad \textit{where} \qquad \widehat{\gamma}_h^{\mathrm{VB}} = \begin{cases} \breve{\gamma}_h^{\mathrm{VB}} & \textit{if } \gamma_h \geq \underline{\gamma}_h^{\mathrm{VB}}, \\ 0 & \textit{otherwise.} \end{cases}$$

Here, the truncation threshold is given by

$$\underline{\gamma}_h^{\mathrm{VB}} = \sigma \sqrt{\frac{(L+M)}{2} + \frac{\sigma^2}{2c_{a_h}^2 c_{b_h}^2} + \sqrt{\left(\frac{(L+M)}{2} + \frac{\sigma^2}{2c_{a_h}^2 c_{b_h}^2}\right)^2 - LM}},$$

and the shrinkage estimator $\breve{\gamma}_h^{\mathrm{VB}}$ *is the second largest real solution of a* quartic *equation.*[6]

With Proposition 18, it is sufficient to obtain the new analytic-form (17) of the shrinkage estimator for proving Theorem 2. However, we give a proof, starting not from Proposition 18 but from Proposition 1. Thanks to the new analytic-form of the shrinkage estimator, our new proof is much more intuitive than the proof given in Nakajima and Sugiyama (2011) and in Nakajima et al. (2013b), for example, in choosing the global solution from two stationary points: the free energy is directly compared in the new proof, while it was shown that one of the stationary points is a saddle point by evaluating the Hessian in Nakajima and Sugiyama (2011).

Proposition 1 states that the VB estimator can be obtained by minimizing the free energy (14) for each singular component separately. Clearly, Eq.(14) is differentiable, and diverges to $F_h \to \infty$ as any variable approaches to any point on the domain boundary. Therefore, any minimizer is stationary point.

The stationary condition of Eq.(14) is given by

$$\widehat{a}_h = \frac{1}{\sigma^2} \gamma_h \widehat{b}_h \sigma_{a_h}^2, \tag{64}$$

$$\sigma_{a_h}^2 = \sigma^2 \left(\widehat{b}_h^2 + L\sigma_{b_h}^2 + \frac{\sigma^2}{c_{a_h}^2}\right)^{-1}, \tag{65}$$

$$\widehat{b}_h = \frac{1}{\sigma^2} \gamma_h \widehat{a}_h \sigma_{b_h}^2, \tag{66}$$

$$\sigma_{b_h}^2 = \sigma^2 \left(\widehat{a}_h^2 + M\sigma_{a_h}^2 + \frac{\sigma^2}{c_{b_h}^2}\right)^{-1}. \tag{67}$$

By using Eqs.(65) and (67), the free energy (14) can be written as

$$F_h = M \log \frac{c_{a_h}^2}{\sigma_{a_h}^2} + L \log \frac{c_{b_h}^2}{\sigma_{b_h}^2} + \frac{\sigma^2}{\sigma_{a_h}^2 \sigma_{b_h}^2} - \frac{2\widehat{a}_h \widehat{b}_h \gamma_h}{\sigma^2} - \left(L + M + \frac{\sigma^2}{c_{a_h}^2 c_{b_h}^2}\right). \tag{68}$$

The stationary condition, Eqs.(64)–(67), implies two possibilities of stationary points.

[6] The quartic equation is omitted, since it is complicated and no longer important.

A.1 Null Stationary Point

If $\widehat{a}_h = 0$ or $\widehat{b}_h = 0$, Eqs.(64) and (66) require that $\widehat{a}_h = 0$ *and* $\widehat{b}_h = 0$. In this case, Eqs.(65) and (67) lead to

$$\sigma_{a_h}^2 = c_{a_h}^2 \left(1 - \frac{L\sigma_{a_h}^2\sigma_{b_h}^2}{\sigma^2}\right), \tag{69}$$

$$\sigma_{b_h}^2 = c_{b_h}^2 \left(1 - \frac{M\sigma_{a_h}^2\sigma_{b_h}^2}{\sigma^2}\right). \tag{70}$$

Multiplying Eqs.(69) and (70), we have

$$\left(1 - \frac{L\sigma_{a_h}^2\sigma_{b_h}^2}{\sigma^2}\right)\left(1 - \frac{M\sigma_{a_h}^2\sigma_{b_h}^2}{\sigma^2}\right) = \frac{\sigma_{a_h}^2\sigma_{b_h}^2}{c_{a_h}^2 c_{b_h}^2}, \tag{71}$$

and therefore

$$\frac{LM}{\sigma^2}\sigma_{a_h}^4\sigma_{b_h}^4 - \left(L + M + \frac{\sigma^2}{c_{a_h}^2 c_{b_h}^2}\right)\sigma_{a_h}^2\sigma_{b_h}^2 + \sigma^2 = 0. \tag{72}$$

Solving the quadratic equation (72) with respect to $\sigma_{a_h}^2\sigma_{b_h}^2$, and checking the signs of $\sigma_{a_h}^2$ and $\sigma_{b_h}^2$, we have the following lemma (the proof is given in Appendix G.1):

Lemma 19 *For any $\gamma_h \geq 0$ and $c_{a_h}^2, c_{b_h}^2, \sigma^2 \in \mathbb{R}_{++}$, the null stationary point given by Eq.*(20) *exists with the following free energy:*

$$F_h^{\mathrm{VB-Null}} = -M\log\left(1 - \frac{L}{\sigma^2}\widehat{\zeta}_h^{\mathrm{VB}}\right) - L\log\left(1 - \frac{M}{\sigma^2}\widehat{\zeta}_h^{\mathrm{VB}}\right) - \frac{LM}{\sigma^2}\widehat{\zeta}_h^{\mathrm{VB}}, \tag{73}$$

$$\textit{where} \quad \widehat{\zeta}_h^{\mathrm{VB}}\left(\equiv \sigma_{a_h}^2\sigma_{b_h}^2\right) = \frac{\sigma^2}{2LM}\left(L + M + \frac{\sigma^2}{c_{a_h}^2 c_{b_h}^2} - \sqrt{\left(L + M + \frac{\sigma^2}{c_{a_h}^2 c_{b_h}^2}\right)^2 - 4LM}\right). \tag{21}$$

A.2 Positive Stationary Point

Assume that $\widehat{a}_h, \widehat{b}_h \neq 0$. In this case, Eqs.(64) and (66) imply that $\widehat{a}_h$ and $\widehat{b}_h$ have the same sign. Define

$$\widehat{\gamma}_h = \widehat{a}_h\widehat{b}_h > 0,$$
$$\widehat{\delta}_h = \frac{\widehat{a}_h}{\widehat{b}_h} > 0.$$

From Eqs.(64) and (66), we have

$$\sigma_{a_h}^2 = \frac{\sigma^2}{\gamma_h}\widehat{\delta}_h, \tag{74}$$

$$\sigma_{b_h}^2 = \frac{\sigma^2}{\gamma_h}\widehat{\delta}_h^{-1}. \tag{75}$$

Substituting Eqs.(74) and (75) into Eqs.(65) and (67) gives

$$\widehat{\delta}_h = \frac{c_{a_h}^2}{\sigma^2}\left(\gamma_h - \widehat{\gamma}_h - \frac{L\sigma^2}{\gamma_h}\right), \tag{76}$$

$$\widehat{\delta}_h^{-1} = \frac{c_{b_h}^2}{\sigma^2}\left(\gamma_h - \widehat{\gamma}_h - \frac{M\sigma^2}{\gamma_h}\right). \tag{77}$$

Multiplying Eqs.(76) and (77), we have

$$\left(\gamma_h - \widehat{\gamma}_h - \frac{L\sigma^2}{\gamma_h}\right)\left(\gamma_h - \widehat{\gamma}_h - \frac{M\sigma^2}{\gamma_h}\right) = \frac{\sigma^4}{c_{a_h}^2 c_{b_h}^2}, \tag{78}$$

and therefore

$$\widehat{\gamma}_h^2 - \left(2\gamma_h - \frac{(L+M)\sigma^2}{\gamma_h}\right)\widehat{\gamma}_h + \left(\gamma_h - \frac{L\sigma^2}{\gamma_h}\right)\left(\gamma_h - \frac{M\sigma^2}{\gamma_h}\right) - \frac{\sigma^4}{c_{a_h}^2 c_{b_h}^2} = 0. \tag{79}$$

By solving the quadratic equation (79) with respect to $\widehat{\gamma}_h$, and checking the signs of $\widehat{\gamma}_h, \widehat{\delta}_h, \sigma_{a_h}^2$ and $\sigma_{b_h}^2$, we have the following lemma (the proof is given in Appendix G.2):

Lemma 20 *If and only if* $\gamma_h > \underline{\gamma}_h^{\mathrm{VB}}$, *where*

$$\underline{\gamma}_h^{\mathrm{VB}} = \sigma\sqrt{\frac{(L+M)}{2} + \frac{\sigma^2}{2c_{a_h}^2 c_{b_h}^2} + \sqrt{\left(\frac{(L+M)}{2} + \frac{\sigma^2}{2c_{a_h}^2 c_{b_h}^2}\right)^2 - LM}}, \tag{16}$$

the positive stationary point given by Eq.(18) *exists with the following free energy:*

$$F_h^{\mathrm{VB-Posi}} = -M\log\left(1 - \left(\frac{\breve{\gamma}_h^{\mathrm{VB}}}{\gamma_h} + \frac{L\sigma^2}{\gamma_h^2}\right)\right) - L\log\left(1 - \left(\frac{\breve{\gamma}_h^{\mathrm{VB}}}{\gamma_h} + \frac{M\sigma^2}{\gamma_h^2}\right)\right) - \frac{\gamma_h^2}{\sigma^2}\left(\frac{\breve{\gamma}_h^{\mathrm{VB}}}{\gamma_h} + \frac{L\sigma^2}{\gamma_h^2}\right)\left(\frac{\breve{\gamma}_h^{\mathrm{VB}}}{\gamma_h} + \frac{M\sigma^2}{\gamma_h^2}\right), \tag{80}$$

where $\breve{\gamma}_h^{\mathrm{VB}} = \gamma_h\left(1 - \frac{\sigma^2}{2\gamma_h^2}\left(M + L + \sqrt{(M-L)^2 + \frac{4\gamma_h^2}{c_{a_h}^2 c_{b_h}^2}}\right)\right)$. (17)

A.3 Useful Relations

Here, we summarize some useful relations between variables, which are used in the subsequent sections. $\widehat{\zeta}_h^{\mathrm{VB}}, \breve{\gamma}_h^{\mathrm{VB}}$, and $\underline{\gamma}_h^{\mathrm{VB}}$, derived from Eqs.(71), (78), and the constant part of Eq.(79), respectively, satisfy the following:

$$\left(1 - \frac{L\widehat{\zeta}_h^{\mathrm{VB}}}{\sigma^2}\right)\left(1 - \frac{M\widehat{\zeta}_h^{\mathrm{VB}}}{\sigma^2}\right) - \frac{\widehat{\zeta}_h^{\mathrm{VB}}}{c_{a_h}^2 c_{b_h}^2} = 0, \tag{81}$$

$$\left(\gamma_h - \breve{\gamma}_h^{\mathrm{VB}} - \frac{L\sigma^2}{\gamma_h}\right)\left(\gamma_h - \breve{\gamma}_h^{\mathrm{VB}} - \frac{M\sigma^2}{\gamma_h}\right) - \frac{\sigma^4}{c_{a_h}^2 c_{b_h}^2} = 0, \tag{82}$$

$$\left(\underline{\gamma}_h^{\mathrm{VB}} - \frac{L\sigma^2}{\underline{\gamma}_h^{\mathrm{VB}}}\right)\left(\underline{\gamma}_h^{\mathrm{VB}} - \frac{M\sigma^2}{\underline{\gamma}_h^{\mathrm{VB}}}\right) - \frac{\sigma^4}{c_{a_h}^2 c_{b_h}^2} = 0. \tag{83}$$

From Eqs.(21) and (16), we find that

$$\underline{\gamma}_h^{\mathrm{VB}} = \sqrt{\left((L+M)\sigma^2 + \frac{\sigma^4}{c_{a_h}^2 c_{b_h}^2}\right) - LM\widehat{\zeta}_h^{\mathrm{VB}}}, \tag{84}$$

which is useful when comparing the free energies of the null and the positive stationary points.

A.4 Free Energy Comparison

Lemma 19 and Lemma 20 imply that, when $\gamma_h \leq \underline{\gamma}_h^{\mathrm{VB}}$, the null stationary point is only the stationary point, and therefore the global solution. When $\gamma_h > \underline{\gamma}_h^{\mathrm{VB}}$, both of the null and the positive stationary points exist, and therefore, identifying the global solution requires to compare the free energies, given by Eqs.(73) and (80), at them.

Given the observed singular value $\gamma_h \geq 0$, we view the free energy as a function of $c_{a_h}^2 c_{b_h}^2$. We also view the threshold $\underline{\gamma}_h^{\mathrm{VB}}$ as a function of $c_{a_h}^2 c_{b_h}^2$. We find from Eq.(16) that $\underline{\gamma}_h^{\mathrm{VB}}$ is decreasing and lower-bounded by $\underline{\gamma}_h^{\mathrm{VB}} > \sqrt{M}\sigma$. Therefore, when $\gamma_h \leq \sqrt{M}\sigma$, $\underline{\gamma}_h^{\mathrm{VB}}$ never gets smaller than γ_h for any $c_{a_h}^2 c_{b_h}^2 > 0$. When $\gamma_h > \sqrt{M}\sigma$ on the other hand, there is a threshold $\underline{c}_{a_h}^2 \underline{c}_{b_h}^2$ such that $\gamma_h > \underline{\gamma}_h^{\mathrm{VB}}$ if $c_{a_h}^2 c_{b_h}^2 > \underline{c}_{a_h}^2 \underline{c}_{b_h}^2$. Eq.(83) implies that the threshold is given by

$$\underline{c}_{a_h}^2 \underline{c}_{b_h}^2 = \frac{\sigma^4}{\gamma_h^2\left(1 - \frac{L\sigma^2}{\gamma_h^2}\right)\left(1 - \frac{M\sigma^2}{\gamma_h^2}\right)}.$$

We have the following lemma (the proof is given in Appendix G.3):

Lemma 21 *For any $\gamma_h \geq 0$ and $c_{a_h}^2 c_{b_h}^2 > 0$, the derivative of the free energy* (73) *at the null stationary point with respect to $c_{a_h}^2 c_{b_h}^2$ is given by*

$$\frac{\partial F_h^{\mathrm{VB-Null}}}{\partial c_{a_h}^2 c_{b_h}^2} = \frac{LM\widehat{\zeta}_h^{\mathrm{VB}}}{\sigma^2 c_{a_h}^2 c_{b_h}^2}. \tag{85}$$

For $\gamma_h > M/\sigma^2$ and $c_{a_h}^2 c_{b_h}^2 > \underline{c}_{a_h}^2 \underline{c}_{b_h}^2$, the derivative of the free energy (80) *at the positive stationary point with respect to $c_{a_h}^2 c_{b_h}^2$ is given by*

$$\frac{\partial F_h^{\mathrm{VB-Posi}}}{\partial c_{a_h}^2 c_{b_h}^2} = \frac{\gamma_h^2}{\sigma^2 c_{a_h}^2 c_{b_h}^2}\left(\frac{(\breve{\gamma}_h^{\mathrm{VB}})^2}{\gamma_h^2} - \left(1 - \frac{(L+M)\sigma^2}{\gamma_h^2}\right)\frac{\breve{\gamma}_h^{\mathrm{VB}}}{\gamma_h} + \frac{LM\sigma^4}{\gamma_h^4}\right). \tag{86}$$

The derivative of the difference is negative, i.e.,

$$\frac{\partial(F_h^{\mathrm{Posi}} - F_h^{\mathrm{Null}})}{\partial c_{a_h}^2 c_{b_h}^2} = -\frac{1}{\sigma^2 c_{a_h}^2 c_{b_h}^2}\left(\gamma_h\left(\gamma_h - \breve{\gamma}_h^{\mathrm{VB}}\right) - (\underline{\gamma}_h^{\mathrm{VB}})^2\right) < 0. \tag{87}$$

It is easy to show that the null stationary point (20) and the positive stationary point (18) coincide with each other at $c_{a_h}^2 c_{b_h}^2 \to \underline{c}_{a_h}^2 \underline{c}_{b_h}^2 + 0$. Therefore,

$$\lim_{c_{a_h}^2 c_{b_h}^2 \to \underline{c}_{a_h}^2 \underline{c}_{b_h}^2 + 0} \left(F_h^{\mathrm{VB-Posi}} - F_h^{\mathrm{VB-Null}} \right) = 0. \tag{88}$$

Eqs.(87) and (88) together imply that

$$F_h^{\mathrm{VB-Posi}} - F_h^{\mathrm{VB-Null}} < 0 \qquad \text{for} \qquad c_{a_h}^2 c_{b_h}^2 > \underline{c}_{a_h}^2 \underline{c}_{b_h}^2,$$

which results in the following lemma:

Lemma 22 *The positive stationary point is the global solution (the global minimizer of the free energy* (14) *for fixed c_{a_h} and c_{b_h}) whenever it exists.*

Figure 8 illustrates the behavior of the free energies.

Combining Lemma 19, Lemma 20, and Lemma 22 completes the proof of of Theorem 2 and Corollary 3. ■

Appendix B. Proof of Theorem 4, Corollary 5, and Corollary 6

The EVB solution was also previously obtained:

Proposition 23 *(Nakajima et al., 2013b) The EVB solution is given by*

$$\widehat{\gamma}_h^{\mathrm{EVB}} = \begin{cases} \breve{\gamma}_h^{\mathrm{VB}} & \text{if } \gamma_h > (\sqrt{L} + \sqrt{M})\sigma \text{ and } F_h \leq 0, \\ 0 & \text{otherwise}, \end{cases}$$

where $\breve{\gamma}_h^{\mathrm{VB}}$ is the VB solution for $c_{a_h}^2 c_{b_h}^2 = \widehat{c}_{a_h}^2 \widehat{c}_{b_h}^2$, and

$$\widehat{c}_{a_h}^2 \widehat{c}_{b_h}^2 = \frac{1}{2LM} \left(\gamma_h^2 - (L+M)\sigma^2 + \sqrt{\left(\gamma_h^2 - (L+M)\sigma^2\right)^2 - 4LM\sigma^4} \right),$$

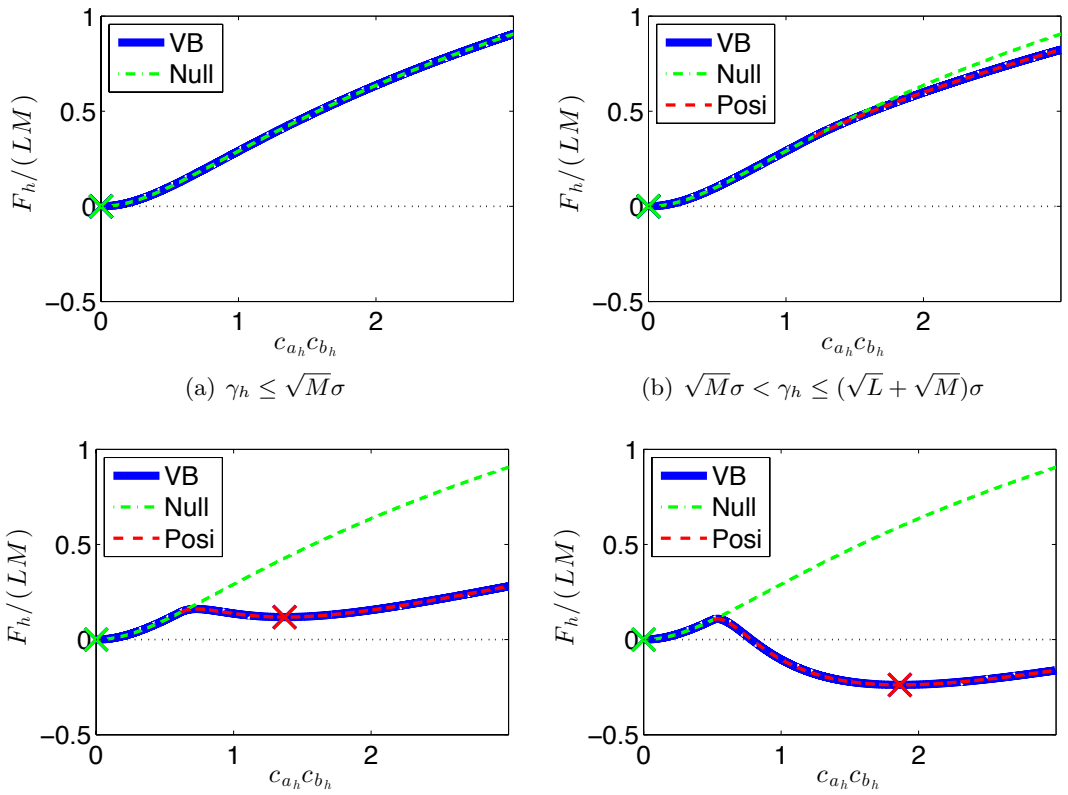
$$F_h = M \log \left(\frac{\gamma_h}{M\sigma^2} \breve{\gamma}_h^{\mathrm{VB}} + 1 \right) + L \log \left(\frac{\gamma_h}{L\sigma^2} \breve{\gamma}_h^{\mathrm{VB}} + 1 \right) + \frac{-2\gamma_h \breve{\gamma}_h^{\mathrm{VB}} + LM \widehat{c}_{a_h}^2 \widehat{c}_{b_h}^2}{\sigma^2}.$$

However, Proposition 23 requires to solve a quartic equation for obtaining $\breve{\gamma}_h^{\mathrm{VB}}$, and moreover, to evaluate the free energy F_h at the obtained $\breve{\gamma}_h^{\mathrm{VB}}$. This obstructs further analysis.

In this appendix, we prove Theorem 4, which provides explicit-forms, (25) and (26), of the EVB threshold $\underline{\gamma}^{\mathrm{EVB}}$ and the EVB shrinkage estimator $\breve{\gamma}_h^{\mathrm{EVB}}$. Without relying on Proposition 23, we can easily obtain Eq.(26) in an intuitive way, by using some of the results obtained in Appendix A. After that, by expressing the free energy F_h with rescaled observation and estimator, we derive Eq.(25).

B.1 EVB Shrinkage Estimator

Eqs.(73) and (80) imply that the free energy does not depend on the ratio c_{a_h}/c_{b_h} between the hyperparameters. Accordingly, we fix the ratio to $c_{a_h}/c_{b_h} = 1$. Lemma 21 allows us to minimize the free energy with respect to $c_{a_h} c_{b_h}$ in a straight-forward way.



(a) $\gamma_h \leq \sqrt{M}\sigma$ (b) $\sqrt{M}\sigma < \gamma_h \leq (\sqrt{L}+\sqrt{M})\sigma$

(c) $(\sqrt{L}+\sqrt{M})\sigma < \gamma_h < \underline{\gamma}^{\mathrm{EVB}}$ (d) $\gamma_h \geq \underline{\gamma}^{\mathrm{EVB}}$

Figure 8: Behavior of the free energies (73) and (80) at the null and the positive stationary points as functions of $c_{a_h}c_{b_h}$, when $L = M = H = 1$ and $\sigma^2 = 1$. The blue curve shows the VB free energy $F_h = \min(F_h^{\mathrm{VB-Null}}, F_h^{\mathrm{VB-Posi}})$ at the global solution, given $c_{a_h}c_{b_h}$. If $\gamma_h \leq \sqrt{M}\sigma$, only the null stationary point exists for any $c_{a_h}c_{b_h} > 0$. Otherwise, the positive stationary point exists for $c_{a_h}c_{b_h} > \underline{c}_{a_h}\underline{c}_{b_h}$, and it is the global minimum whenever it exists. In EVB where $c_{a_h}c_{b_h}$ is also optimized, $c_{a_h}c_{b_h} \to 0$ (indicated by a green cross) is the unique local minimum if $\gamma_h \leq (\sqrt{L}+\sqrt{M})\sigma$. Otherwise, a positive local minimum also exists (indicated by a red cross), and it is the global minimum if and only if $\gamma_h \geq \underline{\gamma}^{\mathrm{EVB}}$.

We see the free energies (73) and (80) at the null and the positive stationary points as function of $c_{a_h}c_{b_h}$ (see Figure 8). We find from Eq.(85) that

$$\frac{\partial F_h^{\mathrm{VB-Null}}}{\partial c_{a_h}^2 c_{b_h}^2} > 0,$$

which implies that the free energy (73) at the null stationary point is increasing. Using Lemma 19, we thus have the following lemma:

Lemma 24 *For any given $\gamma_h \geq 0$ and $\sigma^2 > 0$, the null EVB local solution given by*

$$\widehat{a}_h = 0, \qquad \widehat{b}_h = 0, \qquad \sigma^2_{a_h} = \sqrt{\widehat{\zeta}^{\mathrm{EVB}}}, \qquad \sigma^2_{b_h} = \sqrt{\widehat{\zeta}^{\mathrm{EVB}}}, \qquad c_{a_h} c_{b_h} = \sqrt{\widehat{\zeta}^{\mathrm{EVB}}},$$
$$\textit{where} \qquad \widehat{\zeta}^{\mathrm{EVB}} \to +0,$$

exists with the free energy that converges to

$$F_h^{\mathrm{EVB-Null}} \to +0. \tag{89}$$

When $\gamma_h \geq (\sqrt{L} + \sqrt{M})\sigma$, the derivative (86) of the free energy (80) at the positive stationary point can be further factorized as

$$\frac{\partial F_h^{\mathrm{VB-Posi}}}{\partial c^2_{a_h} c^2_{b_h}} = \frac{\gamma_h}{\sigma^2 c^2_{a_h} c^2_{b_h}} \left(\breve{\gamma}_h^{\mathrm{VB}} - \acute{\gamma}_h\right) \left(\breve{\gamma}_h^{\mathrm{VB}} - \breve{\gamma}_h^{\mathrm{EVB}}\right), \tag{90}$$

$$\text{where} \qquad \acute{\gamma}_h = \frac{\gamma_h}{2} \left(1 - \frac{(L+M)\sigma^2}{\gamma_h^2} - \sqrt{\left(1 - \frac{(L+M)\sigma^2}{\gamma_h^2}\right)^2 - \frac{4LM\sigma^4}{\gamma_h^4}}\right), \tag{91}$$

$$\breve{\gamma}_h^{\mathrm{EVB}} = \frac{\gamma_h}{2} \left(1 - \frac{(L+M)\sigma^2}{\gamma_h^2} + \sqrt{\left(1 - \frac{(L+M)\sigma^2}{\gamma_h^2}\right)^2 - \frac{4LM\sigma^4}{\gamma_h^4}}\right). \tag{26}$$

The VB shrinkage estimator (17) is an increasing function of $c_{a_h} c_{b_h}$ ranging over

$$0 < \breve{\gamma}_h^{\mathrm{VB}} < \gamma_h - \frac{M\sigma^2}{\gamma_h},$$

and both of Eqs.(91) and (26) are in this range, i.e.,

$$0 < \acute{\gamma}_h \leq \breve{\gamma}_h^{\mathrm{EVB}} < \gamma_h - \frac{M\sigma^2}{\gamma_h}.$$

Therefore Eq.(90) leads to the following lemma:

Lemma 25 *If $\gamma_h \leq (\sqrt{L}+\sqrt{M})\sigma$, the free energy $F_h^{\mathrm{VB-Posi}}$ at the positive stationary point is monotonically increasing. Otherwise,*

$$F_h^{\mathrm{VB-Posi}} \textit{ is } \begin{cases} \textit{increasing} & \textit{for} \quad \breve{\gamma}_h^{\mathrm{VB}} < \acute{\gamma}_h, \\ \textit{decreasing} & \textit{for} \quad \acute{\gamma}_h < \breve{\gamma}_h^{\mathrm{VB}} < \breve{\gamma}_h^{\mathrm{EVB}}, \\ \textit{increasing} & \textit{for} \quad \breve{\gamma}_h^{\mathrm{VB}} > \breve{\gamma}_h^{\mathrm{EVB}}, \end{cases}$$

and therefore, minimized at $\breve{\gamma}_h^{\mathrm{VB}} = \breve{\gamma}_h^{\mathrm{EVB}}$.

We can see this behavior of the free energy in Figure 8.

The derivative (86) is zero when $\breve{\gamma}_h^{\mathrm{VB}} = \breve{\gamma}_h^{\mathrm{EVB}}$, which leads to

$$\left(\breve{\gamma}_h^{\mathrm{EVB}} + \frac{L\sigma^2}{\gamma_h}\right) \left(\breve{\gamma}_h^{\mathrm{EVB}} + \frac{M\sigma^2}{\gamma_h}\right) = \gamma_h \breve{\gamma}_h^{\mathrm{EVB}}. \tag{92}$$

Using Eq.(92), we obtain the following lemma (the proof is given in Appendix G.4):

Lemma 26 *If and only if*

$$\gamma_h \geq \underline{\gamma}^{\mathrm{local-EVB}} \equiv (\sqrt{L}+\sqrt{M})\sigma, \tag{28}$$

the positive EVB local solution given by

$$\widehat{a}_h = \pm\sqrt{\breve{\gamma}_h^{\mathrm{EVB}}\widehat{\delta}_h^{\mathrm{EVB}}}, \qquad \widehat{b}_h = \pm\sqrt{\frac{\breve{\gamma}_h^{\mathrm{EVB}}}{\widehat{\delta}_h^{\mathrm{EVB}}}}, \qquad \sigma_{a_h}^2 = \frac{\sigma^2\widehat{\delta}_h^{\mathrm{EVB}}}{\gamma_h}, \qquad \sigma_{b_h}^2 = \frac{\sigma^2}{\gamma_h\widehat{\delta}_h^{\mathrm{EVB}}}, \tag{93}$$

$$c_{a_h}c_{b_h} = \sqrt{\frac{\gamma_h\breve{\gamma}_h^{\mathrm{EVB}}}{LM}}, \qquad \textit{where} \qquad \widehat{\delta}_h^{\mathrm{EVB}} = \sqrt{\frac{M\breve{\gamma}_h^{\mathrm{EVB}}}{L\gamma_h}}\left(1+\frac{L\sigma^2}{\gamma_h\breve{\gamma}_h^{\mathrm{EVB}}}\right), \tag{94}$$

$$\breve{\gamma}_h^{\mathrm{EVB}} = \frac{\gamma_h}{2}\left(1-\frac{(M+L)\sigma^2}{\gamma_h^2}+\sqrt{\left(1-\frac{(M+L)\sigma^2}{\gamma_h^2}\right)^2-\frac{4LM\sigma^4}{\gamma_h^4}}\right), \tag{26}$$

exists with the following free energy:

$$F_h^{\mathrm{EVB-Posi}} = M\log\left(\frac{\gamma_h\breve{\gamma}_h^{\mathrm{EVB}}}{M\sigma^2}+1\right)+L\log\left(\frac{\gamma_h\breve{\gamma}_h^{\mathrm{EVB}}}{L\sigma^2}+1\right)-\frac{\gamma_h\breve{\gamma}_h^{\mathrm{EVB}}}{\sigma^2}. \tag{95}$$

In Figure 8, the positive EVB local solution at $c_{a_h}c_{b_h} = \sqrt{\gamma_h\breve{\gamma}_h^{\mathrm{EVB}}/(LM)}$ is indicated by a red cross if it exists.

B.2 EVB Threshold

Lemma 24 and Lemma 26 state that, if $\gamma_h \leq \underline{\gamma}^{\mathrm{local-EVB}}$, only the null EVB local solution exists, and therefore it is the global EVB solution. Below, assuming that $\gamma_h \geq \underline{\gamma}^{\mathrm{local-EVB}}$, we compare the free energy (89) at the null EVB local solution and the free energy (95) at the positive EVB local solution. Since $F_h^{\mathrm{EVB-Null}} \to +0$, we simply clarify when $F_h^{\mathrm{EVB-Posi}} \leq 0$. Eq.(92) gives

$$\left(\gamma_h\breve{\gamma}_h^{\mathrm{EVB}}+L\sigma^2\right)\left(1+\frac{M\sigma^2}{\gamma_h\breve{\gamma}_h^{\mathrm{EVB}}}\right) = \gamma_h^2. \tag{29}$$

By using Eqs.(26) and (28), we have

$$\begin{aligned}\gamma_h\breve{\gamma}_h^{\mathrm{EVB}} = \frac{1}{2}\Bigg(&\gamma_h^2-\left(\underline{\gamma}^{\mathrm{local-EVB}}\right)^2+2\sqrt{LM}\sigma^2 \\ &+\sqrt{\left(\gamma_h^2-\left(\underline{\gamma}^{\mathrm{local-EVB}}\right)^2\right)\left(\gamma_h^2-\left(\underline{\gamma}^{\mathrm{local-EVB}}\right)^2+4\sqrt{LM}\sigma^2\right)}\Bigg) \\ \geq\ &\sqrt{LM}\sigma^2.\end{aligned} \tag{30}$$

Let

$$\alpha = \frac{L}{M} \qquad (0<\alpha\leq 1), \tag{22}$$

$$x_h = \frac{\gamma_h^2}{M\sigma^2}, \tag{33}$$

$$\tau_h = \frac{\gamma_h \breve{\gamma}_h^{\mathrm{EVB}}}{M\sigma^2}. \tag{34}$$

Eqs.(29) and (26) imply the following mutual relations between x_h and τ_h:

$$x_h \equiv x(\tau_h;\alpha) = (1+\tau_h)\left(1+\frac{\alpha}{\tau_h}\right), \tag{35}$$

$$\tau_h \equiv \tau(x_h;\alpha) = \frac{1}{2}\left(x_h - (1+\alpha) + \sqrt{(x_h-(1+\alpha))^2 - 4\alpha}\right). \tag{36}$$

Eqs.(28) and (30) lead to

$$x_h \geq \underline{x}^{\mathrm{local}} = \frac{(\underline{\gamma}^{\mathrm{local-EVB}})^2}{M\sigma^2} = x(\sqrt{\alpha};\alpha) = (1+\sqrt{\alpha})^2, \tag{37}$$

$$\tau_h \geq \underline{\tau}^{\mathrm{local}} = \sqrt{\alpha}. \tag{38}$$

Then, using

$$\Xi(\tau;\alpha) = \Phi(\tau) + \Phi\left(\frac{\tau}{\alpha}\right), \qquad \text{where} \qquad \Phi(z) = \frac{\log(z+1)}{z} - \frac{1}{2}, \tag{23}$$

we can rewrite Eq.(95) as

$$\begin{aligned} F_h^{\mathrm{EVB-Posi}} &= M\log(\tau_h+1) + L\log\left(\frac{\tau_h}{\alpha}+1\right) - M\tau_h \\ &= M\tau_h \Xi(\tau;\alpha). \end{aligned} \tag{96}$$

The following holds for $\Phi(z)$ (the proof is given in Appendix G.5):

Lemma 27 *$\Phi(z)$ is decreasing for $z > 0$.*

Figure 9 shows $\Phi(z)$. Since $\Phi(z)$ is decreasing, $\Xi(\tau;\alpha)$ is also decreasing with respect to τ. It holds that, for any $0 < \alpha \leq 1$,

$$\lim_{\tau\to 0} \Xi(\tau;\alpha) = 1,$$
$$\lim_{\tau\to\infty} \Xi(\tau;\alpha) = -1.$$

Therefore, $\Xi(\tau;\alpha)$ has a unique zero-cross point $\underline{\tau}$, such that

$$\Xi(\tau;\alpha) \leq 0 \qquad \text{if and only if} \qquad \tau \geq \underline{\tau}. \tag{97}$$

We can prove the following lemma (the proof is given in Appendix G.6):

Lemma 28 *The unique zero-cross point $\underline{\tau}$ of $\Xi(\tau;\alpha)$ lies in the following range:*

$$\sqrt{\alpha} < \underline{\tau} \leq \underline{z}, \tag{27}$$

where $\underline{z} \approx 2.5129$ is the unique zero-cross point of $\Phi(z)$.

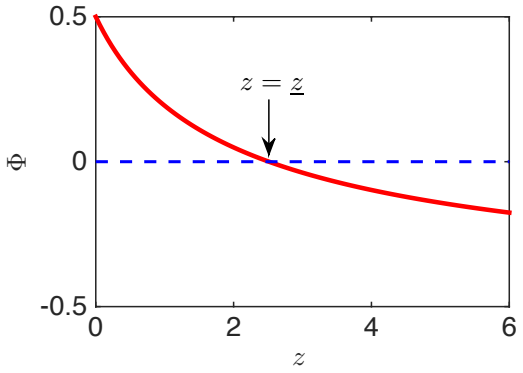


Figure 9: $\Phi(z) = \frac{\log(z+1)}{z} - \frac{1}{2}$. $\underline{z} \approx 2.5129$ is the unique zero cross point, i.e., $\Phi(\underline{z}) = 0$.

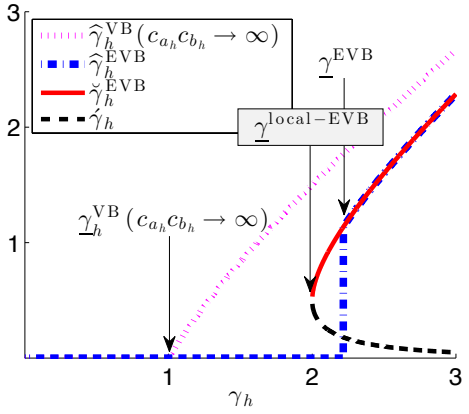


Figure 10: Estimators and thresholds for $L = M = H = 1$ and $\sigma^2 = 1$.

Since Eq.(35) is increasing with respect to τ_h ($> \sqrt{\alpha}$), the thresholding condition $\tau \geq \underline{\tau}$ in Eq.(97) can be expressed in terms of x:

$$\Xi(\tau(x);\alpha) \leq 0 \qquad \text{if and only if} \qquad x \geq \underline{x},$$
$$\text{where} \qquad \underline{x} \equiv x(\underline{\tau};\alpha) = (1+\underline{\tau})\left(1+\frac{\alpha}{\underline{\tau}}\right). \tag{39}$$

Using Eqs.(33) and (96), we have

$$F_h^{\mathrm{EVB-Posi}} \leq 0 \qquad \text{if and only if} \qquad \gamma_h \geq \underline{\gamma}^{\mathrm{EVB}},$$
$$\text{where} \qquad \underline{\gamma}^{\mathrm{EVB}} = \sigma\sqrt{M(1+\underline{\tau})\left(1+\frac{\alpha}{\underline{\tau}}\right)}. \tag{25}$$

Thus, we have the following lemma:

Lemma 29 *The positive EVB local solution is the global EVB solution if and only if* $\gamma_h \geq \underline{\gamma}^{\mathrm{EVB}}$.

Combining Lemma 24, Lemma 26, and Lemma 29 completes the proof of Theorem 4 and Corollary 6. All formulas in Corollary 5 have already been derived. ■

Figure 10 shows estimators and thresholds for $L = M = H = 1$ and $\sigma^2 = 1$. The curves indicate the VB solution $\widehat{\gamma}_h^{\mathrm{VB}}$, given by Eq.(15), the EVB solution $\widehat{\gamma}_h^{\mathrm{EVB}}$, given by Eq.(24), the EVB positive local minimizer $\breve{\gamma}_h^{\mathrm{EVB}}$, given by Eq.(26), and the EVB positive local maximizer $\acute{\gamma}_h$, given by Eq.(91), respectively. The arrows indicate the VB threshold $\underline{\gamma}_h^{\mathrm{VB}}$, given by Eq.(16), the local-EVB threshold $\underline{\gamma}^{\mathrm{local-EVB}}$, given by Eq.(28), and the EVB threshold $\underline{\gamma}^{\mathrm{EVB}}$, given by Eq.(25), respectively.

Appendix C. Proof of Theorem 7

By using Lemma 24 and Lemma 26, the free energy (13) can be written as a function of σ^2:

$$2F = LM\log(2\pi\sigma^2) + \frac{\sum_{h=1}^{L}\gamma_h^2}{\sigma^2} + \sum_{h=1}^{H}\theta\left(\gamma_h > \underline{\gamma}^{\mathrm{EVB}}\right)F_h^{\mathrm{EVB-Posi}}, \tag{98}$$

$$\text{where} \quad F_h^{\mathrm{EVB-Posi}} = M\log\left(\frac{\gamma_h\breve{\gamma}_h^{\mathrm{EVB}}}{M\sigma^2}+1\right) + L\log\left(\frac{\gamma_h\breve{\gamma}_h^{\mathrm{EVB}}}{L\sigma^2}+1\right) - \frac{\gamma_h\breve{\gamma}_h^{\mathrm{EVB}}}{\sigma^2}. \tag{95}$$

By using Eqs.(34) and (36), Eq.(95) can be written as

$$\begin{aligned} F_h^{\mathrm{EVB-Posi}} &= M\log\left(\tau_h+1\right) + L\log\left(\frac{\tau_h}{\alpha}+1\right) - M\tau_h \\ &= M\psi_1(x_h). \end{aligned} \tag{99}$$

Therefore, Eq.(98) is written as

$$\begin{aligned} 2F &= M\left\{\sum_{h=1}^{L}\log\left(\frac{2\pi\gamma_h^2}{M}\right) + \sum_{h=1}^{L}\left(\log\left(\frac{M\sigma^2}{\gamma_h^2}\right)+\frac{\gamma_h^2}{M\sigma^2}\right) + \sum_{h=1}^{H}\theta\left(\gamma_h > \underline{\gamma}^{\mathrm{EVB}}\right)\frac{F_h^{\mathrm{EVB-Posi}}}{M}\right\} \\ &= M\left\{\sum_{h=1}^{L}\log\left(\frac{2\pi\gamma_h^2}{M}\right) + \sum_{h=1}^{L}\psi_0(x_h) + \sum_{h=1}^{H}\theta\left(x_h > \underline{x}\right)\psi_1(x_h)\right\}. \end{aligned}$$

Note that the first term in the curly braces is constant with respect to σ^2. By defining

$$\Omega = \frac{2F}{LM} - \frac{1}{L}\sum_{h=1}^{L}\log\left(\frac{2\pi\gamma_h^2}{M}\right),$$

we obtain Eq.(40), which completes the proof of Theorem 7. ■

Appendix D. Proof of Theorem 8 and Corollary 9

First, we investigate properties of the following functions, which are depicted in Fig. 3:

$$\psi(x) = \psi_0(x) + \theta\left(x > \underline{x}\right)\psi_1(x), \tag{41}$$

$$\psi_0(x) = x - \log x, \tag{42}$$

$$\text{where} \quad \psi_1(x) = \log\left(\tau(x;\alpha)+1\right) + \alpha\log\left(\frac{\tau(x;\alpha)}{\alpha}+1\right) - \tau(x;\alpha). \tag{43}$$

They have nice properties (the proof is given in Appendix G.7):

Lemma 30 *The following hold for $x > 0$: $\psi_0(x)$ is differentiable and strictly convex; $\psi(x)$ is continuous and strictly quasi-convex; $\psi(x)$ is differentiable except $x = \underline{x}$, at which $\psi(x)$ has a discontinuously decreasing derivative, i.e., $\lim_{x\to\underline{x}-0}\partial\psi/\partial x > \lim_{x\to\underline{x}+0}\partial\psi/\partial x$; Both of $\psi_0(x)$ and $\psi(x)$ are minimized at $x = 1$. For $x > \underline{x}$, $\psi_1(x)$ is negative and decreasing.*

Lemma 30 implies that our objective

$$\Omega(\sigma^{-2}) = \frac{1}{L}\left(\sum_{h=1}^{H}\psi\left(\frac{\gamma_h^2}{M\sigma^2}\right) + \sum_{h=H+1}^{L}\psi_0\left(\frac{\gamma_h^2}{M\sigma^2}\right)\right) \tag{40}$$

is a sum of quasi-convex functions with respect to σ^{-2}. Therefore, its minimizer can be bounded by the smallest and the largest ones of the minimizers of each quasi-convex function (the proof is given in Appendix G.8):

Lemma 31 *$\Omega(\sigma^{-2})$ has at least one global minimizer, and any of its local minimizers is bounded as*

$$\frac{M}{\gamma_1^2} \leq \widehat{\sigma}^{-2} \leq \frac{M}{\gamma_L^2}.$$

$\Omega(\sigma^{-2})$ has at most H non-differentiable points, which come from the non-differentiable point $x = \underline{x}$ of $\psi(x)$. The values

$$\underline{\sigma}_h^{-2} = \begin{cases} 0 & \text{for } h = 0, \\ \frac{M\underline{x}}{\gamma_h^2} & \text{for } h = 1, \ldots, L, \\ \infty & \text{for } h = L+1, \end{cases} \tag{100}$$

defined in Eq.(45), for $h = 1, \ldots, H$ actually correspond to those points.

Lemma 30 states that, at $x = \underline{x}$, $\psi(x)$ has a discontinuously decreasing derivative and neither $\psi_0(x)$ nor $\psi(x)$ has discontinuously increasing derivative at any point. Therefore, none of those non-differentiable points can be local minimum. Consequently, we have the following lemma:

Lemma 32 *$\Omega(\sigma^{-2})$ has no local minimizer at $\sigma^{-2} = \underline{\sigma}_h^{-2}$ for $h = 1, \ldots, H$, and therefore, any of its local minimizer is stationary point.*

Then, Theorem 4 leads to the following lemma:

Lemma 33 *The estimated rank is $\widehat{H} = h$, if and only if the inverse noise variance estimator lies in the range*

$$\widehat{\sigma}^{-2} \in \mathbb{B}_h \equiv \left\{\sigma^{-2}; \underline{\sigma}_h^{-2} < \sigma^{-2} < \underline{\sigma}_{h+1}^{-2}\right\}.$$

Figure 11 shows quasi-convex functions $\{\psi(\gamma_h^2\sigma^{-2}/M)\}_{h=1}^{H}$ and their sum $\Omega(\sigma^{-2})$ in two example cases for $H = L$. In the left case, the inverse noise variance estimator $\widehat{\sigma}^{-2}$ is smaller than the inverse threshold $\underline{\sigma}_1^{-2}$ for the largest singular value, and therefore, no EVB estimator $\widehat{\gamma}_h$ is positive, i.e., $\widehat{H} = 0$. In the right case, it holds that $\underline{\sigma}_1^{-2} < \widehat{\sigma}^{-2} < \underline{\sigma}_2^{-2}$, and therefore, $\widehat{\gamma}_1$ is positive and the others are zero, i.e., $\widehat{H} = 1$.

We have the following lemma (the proof is given in Appendix G.9):

Lemma 34 *The derivative of $\Omega(\sigma^{-2})$ is given by*

$$\Theta \equiv \frac{\partial\Omega}{\partial\sigma^{-2}} = -\sigma^2 + \frac{\sum_{h=1}^{\widehat{H}}\gamma_h\left(\gamma_h - \breve{\gamma}_h^{\mathrm{EVB}}\right) + \sum_{h=\widehat{H}+1}^{L}\gamma_h^2}{LM}, \tag{101}$$

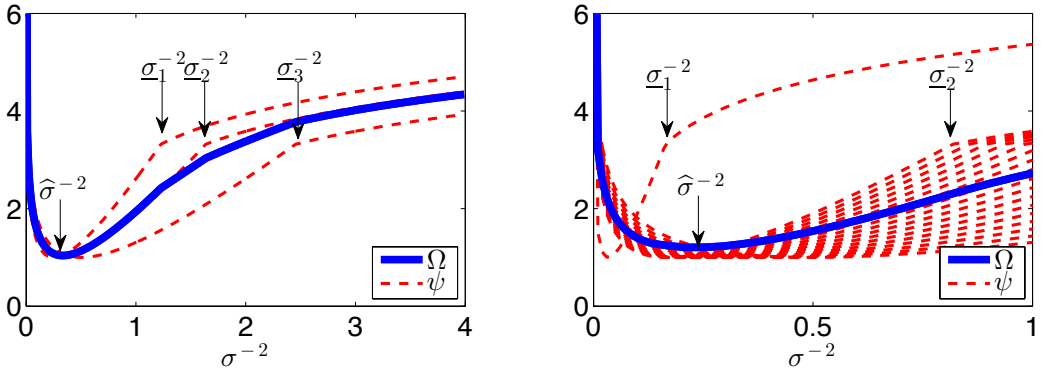


Figure 11: $\{\psi(\gamma_h^2\sigma^{-2}/M)\}_{h=1}^H$ and $\Omega(\sigma^{-2})$ in two example cases for $H = L$. (Left) The case when $\gamma_h^2/M = 4, 3, 2$ for $h = 1, 2, 3$. (Right) The case when $\gamma_1^2/M = 30$, $\gamma_h^2/M = 6.0, 5.75, 5.5, \ldots, 2.0$ for $h = 2, \ldots, 18$.

where $\widehat{H}$ *is a function of* σ^{-2} *defined by*

$$\widehat{H} = \widehat{H}(\sigma^{-2}) = h \qquad \textit{if} \qquad \sigma^{-2} \in \mathbb{B}_h. \tag{102}$$

Note that Eq.(101) involves the shrinkage estimator $\breve{\gamma}_h^{\mathrm{EVB}}$, which is a function of σ^{-2} (see Eq.(26)). For each hypothetical $\widehat{H}$, the solutions of the equation

$$\Theta = 0 \tag{103}$$

lying in $\sigma^{-2} \in \mathbb{B}_{\widehat{H}}$ are stationary points, and hence candidates for the global minimum. If we can solve Eq.(103) for all $\widehat{H} = 1, \ldots, H$, we can obtain the global solution by evaluating the objective (40) at each obtained stationary points. However, solving Eq.(103) is difficult unless $\widehat{H}$ is small (it is easy to derive a closed-form solution for $\widehat{H} = 0, 1$). Based on Lemma 34, we will obtain tighter bounds than Lemma 31.

Since

$$\gamma_h - \breve{\gamma}_h^{\mathrm{EVB}} > 0,$$

Eq.(101) is upper-bounded by

$$\Theta \leq -\sigma^2 + \sum_{h=1}^{L} \frac{\gamma_h^2}{LM},$$

which leads to the upper-bound given in Eq.(44). Actually, if

$$\left(\sum_{h=1}^{L} \frac{\gamma_h^2}{LM}\right)^{-1} \in \mathbb{B}_0,$$

then

$$\widehat{H} = 0,$$

$$\widehat{\sigma}^2 = \sum_{h=1}^{L} \frac{\gamma_h^2}{LM},$$

is a local minimum.

The following lemma is easily obtained from Eq.(26) by using $z_1 < \sqrt{z_1^2 - z_2^2} < z_1 - z_2$ for $z_1 > z_2 > 0$:

Lemma 35 *For $\gamma_h \geq \underline{\gamma}^{\mathrm{EVB}}$, the EVB shrinkage estimator* (26) *can be bounded as follows:*

$$\gamma_h - \frac{(\sqrt{M}+\sqrt{L})^2\sigma^2}{\gamma_h} < \breve{\gamma}_h^{\mathrm{EVB}} < \gamma_h - \frac{(M+L)\sigma^2}{\gamma_h}.$$

This lemma is important for our analysis, because it allows us to bound the most complicated part of Eq.(101) by terms independent of γ_h, i.e.,

$$(M+L)\sigma^2 < \gamma_h \left(\gamma_h - \breve{\gamma}_h^{\mathrm{EVB}}\right) < (\sqrt{M}+\sqrt{L})^2\sigma^2. \tag{104}$$

Using Eq.(104), we obtain the following lemma (the proof is given in Appendix G.10):

Lemma 36 *Any local minimizer exists in $\sigma^{-2} \in \mathbb{B}_{\widehat{H}}$ such that*

$$\widehat{H} < \frac{L}{1+\alpha},$$

and the following holds for any local minimizer lying in $\sigma^{-2} \in \mathbb{B}_{\widehat{H}}$:

$$\widehat{\sigma}^2 \geq \frac{\sum_{h=\widehat{H}+1}^{L} \gamma_h^2}{LM - \widehat{H}(M+L)}.$$

It holds that

$$\frac{\sum_{h=\widehat{H}+1}^{L} \gamma_h^2}{LM - \widehat{H}(M+L)} \geq \frac{\sum_{h=\widehat{H}+1}^{L} \gamma_h^2}{M(L-\widehat{H})}, \tag{105}$$

of which the right-hand side is decreasing with respect to $\widehat{H}$. Combining Lemma 31, Lemma 32, Lemma 33, Lemma 36, and Eq.(105) completes the proof of Theorem 8. Corollary 9 is easily obtained from Lemma 32 and Lemma 34.

■

Appendix E. Proof of Theorem 13 and Corollary 14

In the large-scale limit, we can substitute the expectation $\langle f(y) \rangle_{p(y)}$ for the summation $L^{-1}\sum_{h=1}^{L} f(y_h)$. We can also substitute the MP distribution $p^{\mathrm{MP}}(y)$ for $p(y)$ in the expectation, since the contribution from the H^* signal components converges to zero. Accordingly, our objective (40) converges to

$$\Omega(\sigma^{-2}) \to \Omega^{\mathrm{LSL}}(\sigma^{-2}) \equiv \int_{\kappa}^{\overline{y}} \psi\left(\sigma^{*2}\sigma^{-2}y\right) p^{\mathrm{MP}}(y)dy + \int_{\underline{y}}^{\kappa} \psi_0\left(\sigma^{*2}\sigma^{-2}y\right) p^{\mathrm{MP}}(y)dy$$

$$= \Omega^{\mathrm{LSL-Full}}(\sigma^{-2}) - \int_{\max(\underline{x}\sigma^2/\sigma^{*2}, \underline{y})}^{\kappa} \psi_1\left(\sigma^{*2}\sigma^{-2}y\right) p^{\mathrm{MP}}(y)dy, \quad (106)$$

$$\text{where} \qquad \Omega^{\mathrm{LSL-Full}}(\sigma^{-2}) \equiv \int_{\underline{y}}^{\overline{y}} \psi\left(\sigma^{*2}\sigma^{-2}y\right) p^{\mathrm{MP}}(y)dy, \quad (107)$$

and κ is a constant satisfying

$$\frac{H}{L} = \int_{\kappa}^{\overline{y}} p^{\mathrm{MP}}(y)dy \qquad (\underline{y} \le \kappa \le \overline{y}).$$

Note that $\underline{x}$, $\underline{y}$, and $\overline{y}$ are defined by Eqs.(39) and (48), and it holds that

$$\underline{x} > \overline{y}. \quad (108)$$

We first investigate Eq.(107), which corresponds to the objective for the full-rank $H = L$ model. Let

$$s = \log(\sigma^{-2}),$$
$$t = \log y \qquad \left(dt = \tfrac{1}{y}dy\right).$$

Then, Eq.(107) is written as a convolution:

$$\widetilde{\Omega}^{\mathrm{LSL-Full}}(s) \equiv \Omega^{\mathrm{LSL-Full}}(e^s) = \int \psi\left(\sigma^{*2}e^{s+t}\right) e^t p^{\mathrm{MP}}(e^t)dt$$
$$= \int \widetilde{\psi}(s+t)p^{\mathrm{LSMP}}(t)dt,$$

where

$$\widetilde{\psi}(s) = \psi(\sigma^{*2}e^s),$$
$$p^{\mathrm{LSMP}}(t) = e^t p^{\mathrm{MP}}(e^t)$$
$$= \frac{\sqrt{(e^t - \underline{y})(\overline{y} - e^t)}}{2\pi\alpha}\theta(\underline{y} < e^t < \overline{y}). \quad (109)$$

Since Lemma 30 states that $\psi(x)$ is quasi-convex, its composition $\widetilde{\psi}(s)$ with the non-decreasing function $\sigma^{*2}e^s$ is also quasi-convex.

The following holds for $p^{\mathrm{LSMP}}(t)$, which we call a log-scaled MP (LSMP) distribution (the proof is given in Appendix G.11):

Lemma 37 *The LSMP distribution* (109) *is log-concave.*

Lemma 37 and Proposition 12 imply that $\widetilde{\Omega}^{\mathrm{LSL-Full}}(s)$ is quasi-convex, and therefore, its composition $\Omega^{\mathrm{LSL-Full}}(\sigma^{-2})$ with the non-decreasing function $\log(\sigma^{-2})$ is quasi-convex. The minimizer of $\Omega^{\mathrm{LSL-Full}}(\sigma^{-2})$ can be found by evaluating the derivative Θ, given by Eq.(101), in the large-scale limit:

$$\Theta^{\mathrm{Full}} \to \Theta^{\mathrm{LSL-Full}} = -\sigma^2 + \sigma^{*2}\int_{\underline{y}}^{\overline{y}} y \cdot p^{\mathrm{MP}}(y)dy - \int_{\underline{x}\sigma^2/\sigma^{*2}}^{\overline{y}} \tau(\sigma^{*2}\sigma^{-2}y; \alpha)p^{\mathrm{MP}}(y)dy. \quad (110)$$

Here, we used Eqs.(34) and (36). In the range

$$0 < \sigma^{-2} < \frac{\underline{x}\sigma^{*-2}}{\overline{y}} \qquad \left(i.e., \quad \frac{\underline{x}\sigma^2}{\sigma^{*2}} > \overline{y}\right), \tag{111}$$

the third term in Eq.(110) is zero. Therefore, Eq.(110) is increasing with respect to σ^{-2}, and zero when

$$\sigma^2 = \sigma^{*2} \int_{\underline{y}}^{\overline{y}} y \cdot p^{\mathrm{MP}}(y) dy = \sigma^{*2}.$$

Accordingly, $\Omega^{\mathrm{LSL-Full}}(\sigma^{-2})$ is strictly convex in the range (111). Eq.(108) implies that the point $\sigma^{-2} = \sigma^{*-2}$ is contained in the region (111), and therefore, it is a local minimum of $\Omega^{\mathrm{LSL-Full}}(\sigma^{-2})$. Combined with the quasi-convexity of $\Omega^{\mathrm{LSL-Full}}(\sigma^{-2})$, we have the following lemma:

Lemma 38 *The objective $\Omega^{\mathrm{LSL-Full}}(\sigma^{-2})$ for the full rank model $H = L$ in the large-scale limit is quasi-convex with its minimizer at $\sigma^{-2} = \sigma^{*-2}$. It is strictly convex in the range* (111).

For any κ ($\underline{y} < \kappa < \overline{y}$), the second term in Eq.(106) is zero in the range (111), which includes its minimizer at $\sigma^{-2} = \sigma^{*-2}$. Since Lemma 30 states that $\psi_1(x)$ is decreasing for $x > \underline{x}$, the second term in Eq.(106) is non-decreasing in the region where

$$\left(\sigma^{*-2} <\right) \frac{\underline{x}\sigma^{*-2}}{\overline{y}} \leq \sigma^{-2} < \infty.$$

Therefore, the quasi-convexity of $\Omega^{\mathrm{LSL-Full}}$ is inherited to Ω^{LSL}:

Lemma 39 *The objective $\Omega^{\mathrm{LSL}}(\sigma^{-2})$ for noise variance estimation in the large-scale limit is quasi-convex with its minimizer at $\sigma^{-2} = \sigma^{*-2}$. $\Omega^{\mathrm{LSL}}(\sigma^{-2})$ is strictly convex in the range* (111).

Thus, we have proved that EVB accurately estimates the noise variance in the large-scale limit:

$$\widehat{\sigma}^{2\,\mathrm{EVB}} = \sigma^{*2}.$$

Assume that

$$\nu^*_{H^*} > \sqrt{\alpha}. \tag{51}$$

Then, Proposition 11 guarantees that, in the large-scale limit, it holds that

$$\frac{\gamma^2_{H^*}}{M\sigma^{*2}} \equiv y_{H^*} = (1 + \nu^*_{H^*}) \left(1 + \frac{\alpha}{\nu^*_{H^*}}\right), \tag{112}$$

$$\frac{\gamma^2_{H^*+1}}{M\sigma^{*2}} \equiv y_{H^*+1} = \overline{y} = (1 + \sqrt{\alpha})^2. \tag{113}$$

The EVB threshold is given by

$$\frac{(\underline{\gamma}^{\mathrm{EVB}})^2}{M\widehat{\sigma}^{2\,\mathrm{EVB}}} \equiv \underline{x} = (1+\underline{\tau})\left(1+\frac{\alpha}{\underline{\tau}}\right). \tag{39}$$

Since Lemma 39 states that $\widehat{\sigma}^{2\,\mathrm{EVB}} = \sigma^{*2}$, comparing Eqs.(112) and (113) with Eq.(39) results in the following lemma:

Lemma 40 *It almost surely holds that*

$$\begin{aligned} \gamma_{H^*} \geq \underline{\gamma}^{\mathrm{EVB}} \qquad & \textit{if and only if} \qquad \nu^*_{H^*} \geq \underline{\tau}, \\ \gamma_{H^*+1} < \underline{\gamma}^{\mathrm{EVB}} \qquad & \textit{for any} \qquad \{\nu^*_h\}. \end{aligned}$$

This completes the proof of Theorem 13. Comparing Eqs.(35) and (49) under Lemma 39 and Lemma 40 proves Corollary 14. ■

Appendix F. Proof of Theorem 15 and Corollary 16

We regroup the terms in Eq.(40) as follows:

$$\Omega(\sigma^{-2}) = \Omega_1(\sigma^{-2}) + \Omega_0(\sigma^{-2}), \tag{114}$$

where

$$\Omega_1(\sigma^{-2}) = \frac{1}{H^*}\sum_{h=1}^{H^*} \psi\left(\frac{\gamma_h^2}{M}\sigma^{-2}\right), \tag{115}$$

$$\Omega_0(\sigma^{-2}) = \frac{1}{L-H^*}\left(\sum_{h=H^*+1}^{H} \psi\left(\frac{\gamma_h^2}{M}\sigma^{-2}\right) + \sum_{h=H+1}^{L} \psi_0\left(\frac{\gamma_h^2}{M}\sigma^{-2}\right)\right). \tag{116}$$

Below, assuming that

$$p(y) = p^{\mathrm{SC}}(y), \tag{54}$$

and

$$y_{H^*} > \overline{y}, \tag{117}$$

we derive a sufficient condition for any local minimizer to lie only in $\sigma^{-2} \in \mathbb{B}_{H^*}$, with which Lemma 33 proves the theorem.

Under the assumption (54) and the condition (117), $\Omega_0(\sigma^{-2})$, defined by Eq.(116), is equivalent to the objective $\Omega^{LSL}(\sigma^{-2})$ in the large-scale limit. Using Lemma 39, and noting that

$$\underline{\sigma}^{-2}_{H^*+1} = \frac{M\underline{x}}{\gamma_{H^*+1}}^2 = \frac{\underline{x}\sigma^{*-2}}{\overline{y}} > \sigma^{*-2}, \tag{118}$$

we have the following lemma:

Lemma 41 *$\Omega_0(\sigma^{-2})$ is quasi-convex with its minimizer at*

$$\sigma^{-2} = \sigma^{*-2}.$$

$\Omega_0(\sigma^{-2})$ is strictly convex in the range

$$0 < \sigma^{-2} < \underline{\sigma}_{H^*+1}^{-2}.$$

Using Lemma 41 and the strict quasi-convexity of $\psi(x)$, we can deduce the following lemma (the proof is given in Appendix G.12):

Lemma 42 *$\Omega(\sigma^{-2})$ is non-decreasing (increasing if $\xi > 0$) in the range $\underline{\sigma}_{H^*+1}^2 < \sigma^{-2} < \infty$.*

Using the bounds given by Eq.(104) and Lemma 41, we also obtain the following lemma (the proof is given in Appendix G.13):

Lemma 43 *$\Omega(\sigma^{-2})$ is increasing at $\sigma^{-2} = \underline{\sigma}_{H^*+1}^2 - 0$. It is decreasing at $\sigma^{-2} = \underline{\sigma}_{H^*}^2 + 0$ if the following hold:*

$$\xi < \frac{1}{(1+\sqrt{\alpha})^2}, \tag{119}$$

$$y_{H^*} > \frac{\underline{x}(1-\xi)}{1-\xi(1+\sqrt{\alpha})^2}. \tag{120}$$

Finally, we obtain the following lemma (the proof is given in Appendix G.14):

Lemma 44 *$\Omega(\sigma^{-2})$ is decreasing in the range $0 < \sigma^{-2} < \underline{\sigma}_{H^*}^2$ if the following hold:*

$$\xi < \frac{1}{\underline{x}}, \tag{121}$$

$$y_{H^*} > \frac{\underline{x}(1-\xi)}{1-\underline{x}\xi}. \tag{122}$$

Lemma 42, Lemma 43, and Lemma 44 together state that, if all the conditions (117), (119)–(122) hold, at least one local minimum exists in the correct range $\sigma^{-2} \in \mathbb{B}_{H^*}$, and no local minimum (no stationary point if $\xi > 0$) exists outside the correct range. Therefore, we can estimate the correct rank $\widehat{H}^{\mathrm{EVB}} = H^*$ by using a local search algorithm for noise variance estimation. Choosing the tightest conditions, we have the following lemma:

Lemma 45 *$\Omega(\sigma^{-2})$ has a global minimum in $\sigma^{-2} \in \mathbb{B}_{H^*}$, and no local minimum (no stationary point if $\xi > 0$) outside $\mathbb{B}_{H^*}$, if the following hold:*

$$\xi < \frac{1}{\underline{x}},$$

$$y_{H^*} = \frac{\gamma_{H^*}^2}{M\sigma^{*2}} > \frac{\underline{x}(1-\xi)}{1-\underline{x}\xi}. \tag{123}$$

Using Eq.(49), Eq.(123) can be written with the *true* signal amplitude as follows:

$$(1+\nu^*_{H^*})\left(1+\frac{\alpha}{\nu^*_{H^*}}\right)-\frac{\underline{x}(1-\xi)}{1-\underline{x}\xi}>0.$$

The left-hand side can be factorized as follows:

$$\frac{1}{\nu^*_{H^*}}\left(\nu^*_{H^*}-\frac{\left(\frac{\underline{x}(1-\xi)}{1-\underline{x}\xi}-(1+\alpha)\right)+\sqrt{\left(\frac{\underline{x}(1-\xi)}{1-\underline{x}\xi}-(1+\alpha)\right)^2-4\alpha}}{2}\right)$$
$$\cdot\left(\nu^*_{H^*}-\frac{\left(\frac{\underline{x}(1-\xi)}{1-\underline{x}\xi}-(1+\alpha)\right)-\sqrt{\left(\frac{\underline{x}(1-\xi)}{1-\underline{x}\xi}-(1+\alpha)\right)^2-4\alpha}}{2}\right)>0. \tag{124}$$

When Eq.(51) holds, the last factor in the left-hand side in Eq.(124) is positive. Therefore, we have the following condition:

$$\begin{aligned}\nu^*_{H^*}&>\frac{\left(\frac{\underline{x}(1-\xi)}{1-\underline{x}\xi}-(1+\alpha)\right)+\sqrt{\left(\frac{\underline{x}(1-\xi)}{1-\underline{x}\xi}-(1+\alpha)\right)^2-4\alpha}}{2}\\&=\frac{\left(\frac{\underline{x}-1}{1-\underline{x}\xi}-\alpha\right)+\sqrt{\left(\frac{\underline{x}-1}{1-\underline{x}\xi}-\alpha\right)^2-4\alpha}}{2}.\end{aligned} \tag{125}$$

Lemma 45 with the condition (123) replaced with the condition (125) leads to Theorem 15 and Corollary 16. ■

Appendix G. Proof of Lemmas

Here, we give proofs of the lemmas used in Appendices.

G.1 Proof of Lemma 19

Eq.(72) has two positive real solutions:

$$\sigma^2_{a_h}\sigma^2_{b_h}=\frac{\sigma^2}{2LM}\left(L+M+\frac{\sigma^2}{c^2_{a_h}c^2_{b_h}}\pm\sqrt{\left(L+M+\frac{\sigma^2}{c^2_{a_h}c^2_{b_h}}\right)^2-4LM}\right).$$

The larger solution (with the plus sign) is decreasing with respect to $c^2_{a_h}c^2_{b_h}$, and lower-bounded as $\sigma^2_{a_h}\sigma^2_{b_h}>\sigma^2/L$. The smaller solution (with the minus sign) is increasing with respect to $c^2_{a_h}c^2_{b_h}$, and upper-bounded as $\sigma^2_{a_h}\sigma^2_{b_h}<\sigma^2/M$.

For $\sigma^2_{a_h}$ and $\sigma^2_{b_h}$ to be positive, Eqs.(69) and (70) require that

$$\sigma^2_{a_h}\sigma^2_{b_h}<\frac{\sigma^2}{M},$$

which is violated by the larger solution, while satisfied by the smaller solution. With the smaller solution (21), Eqs.(69) and (70) give the stationary point given by (20).

Using Eq.(72), we can easily derive Eq.(73) from Eq.(68), which completes the proof of Lemma 19. ■

G.2 Proof of Lemma 20

Since $\widehat{\delta} > 0$, Eqs.(76) and (77) require that

$$\widehat{\gamma}_h < \gamma_h - \frac{M\sigma^2}{\gamma_h}, \tag{126}$$

and therefore, the positive stationary point exists only when

$$\gamma_h > \sqrt{M}\sigma. \tag{127}$$

Below, we assume that Eq.(127) holds.

Eq.(79) has two solutions:

$$\widehat{\gamma}_h = \frac{1}{2}\left(2\gamma_h - \frac{(L+M)\sigma^2}{\gamma_h} \pm \sqrt{\left(\frac{(M-L)\sigma^2}{\gamma_h}\right)^2 + \frac{4\sigma^4}{c_{a_h}^2 c_{b_h}^2}}\right).$$

The larger solution with the plus sign is positive, decreasing with respect to $c_{a_h}^2 c_{b_h}^2$, and lower-bounded as $\widehat{\gamma}_h > \gamma_h - L\sigma^2/\gamma_h$, which violates the condition (126).

The smaller solution, Eq.(17), with the minus sign is positive if the intercept of the left-hand side in Eq.(79) is positive, i.e.,

$$\left(\gamma_h - \frac{L\sigma^2}{\gamma_h}\right)\left(\gamma_h - \frac{M\sigma^2}{\gamma_h}\right) - \frac{\sigma^4}{c_{a_h}^2 c_{b_h}^2} > 0. \tag{128}$$

From the condition (128), we obtain the threshold (16) for the existence of the positive stationary point. Note that $\underline{\gamma}_h^{\mathrm{VB}} > \sqrt{M}\sigma$, and therefore, Eq.(127) holds whenever $\gamma_h > \underline{\gamma}_h^{\mathrm{VB}}$.

Assume that $\gamma_h > \underline{\gamma}_h^{\mathrm{VB}}$. Then, with the solution (17), $\widehat{\delta}_h$, given by Eq.(76), and $\sigma_{a_h}^2$ and $\sigma_{b_h}^2$, given by Eqs.(74) and (75), are all positive. Thus, we obtain the positive stationary point (18).

Substituting Eqs.(74) and (75), and then Eqs.(76) and (77), into the free energy (68), we have

$$F_h^{\mathrm{VB-Posi}} = -M\log\left(1 - \frac{\breve{\gamma}_h^{\mathrm{VB}}}{\gamma_h} - \frac{L\sigma^2}{\gamma_h^2}\right) - L\log\left(1 - \frac{\breve{\gamma}_h^{\mathrm{VB}}}{\gamma_h} - \frac{M\sigma^2}{\gamma_h^2}\right) + \frac{-2\gamma_h\breve{\gamma}_h^{\mathrm{VB}}}{\sigma^2} + \frac{\gamma_h^2}{\sigma^2} - \left(L + M + \frac{\sigma^2}{c_{a_h}^2 c_{b_h}^2}\right). \tag{129}$$

Using Eq.(78), we can eliminate the direct dependency on $c_{a_h}^2 c_{b_h}^2$, and express the free energy (129) as a function of $\breve{\gamma}_h^{\mathrm{VB}}$. This results in Eq.(80), and completes the proof of Lemma 20. ■

G.3 Proof of Lemma 21

By differentiating Eqs.(73), (21), (80), and (17), we have

$$\frac{\partial F_h^{\mathrm{VB-Null}}}{\partial \widehat{\zeta}_h^{\mathrm{VB}}} = \frac{LM}{\sigma^2\left(1-\frac{L}{\sigma^2}\widehat{\zeta}_h^{\mathrm{VB}}\right)} + \frac{LM}{\sigma^2\left(1-\frac{M}{\sigma^2}\widehat{\zeta}_h^{\mathrm{VB}}\right)} - \frac{LM}{\sigma^2}$$
$$= \frac{LMc_{a_h}^2c_{b_h}^2\left(1+\frac{\sqrt{LM}}{\sigma^2}\widehat{\zeta}_h^{\mathrm{VB}}\right)\left(1-\frac{\sqrt{LM}}{\sigma^2}\widehat{\zeta}_h^{\mathrm{VB}}\right)}{\sigma^2\widehat{\zeta}_h^{\mathrm{VB}}}, \tag{130}$$

$$\frac{\partial \widehat{\zeta}_h^{\mathrm{VB}}}{\partial c_{a_h}^2c_{b_h}^2} = \frac{\sigma^2}{2LM}\left(-\frac{\sigma^2}{c_{a_h}^4c_{b_h}^4} + \frac{2\sigma^2\left(L+M+\frac{\sigma^2}{c_{a_h}^2c_{b_h}^2}\right)}{2c_{a_h}^4c_{b_h}^4\sqrt{\left(L+M+\frac{\sigma^2}{c_{a_h}^2c_{b_h}^2}\right)^2-4LM}}\right)$$
$$= \frac{1}{c_{a_h}^4c_{b_h}^4}\left(\frac{(\widehat{\zeta}_h^{\mathrm{VB}})^2}{\left(1-\frac{\sqrt{LM}\widehat{\zeta}_h^{\mathrm{VB}}}{\sigma^2}\right)\left(1+\frac{\sqrt{LM}\widehat{\zeta}_h^{\mathrm{VB}}}{\sigma^2}\right)}\right), \tag{131}$$

$$\frac{\partial F_h^{\mathrm{VB-Posi}}}{\partial \breve{\gamma}_h^{\mathrm{VB}}} = \frac{M}{\gamma_h\left(1-\left(\frac{\breve{\gamma}_h^{\mathrm{VB}}}{\gamma_h}+\frac{L\sigma^2}{\gamma_h^2}\right)\right)} + \frac{L}{\gamma_h\left(1-\left(\frac{\breve{\gamma}_h^{\mathrm{VB}}}{\gamma_h}+\frac{M\sigma^2}{\gamma_h^2}\right)\right)} - \frac{\gamma_h}{\sigma^2}\left(\frac{2\breve{\gamma}_h^{\mathrm{VB}}}{\gamma_h}+\frac{(L+M)\sigma^2}{\gamma_h^2}\right)$$
$$= \frac{2c_{a_h}^2c_{b_h}^2\gamma_h^3\left(1-\left(\frac{\breve{\gamma}_h^{\mathrm{VB}}}{\gamma_h}+\frac{(L+M)\sigma^2}{2\gamma_h^2}\right)\right)\left(\frac{(\breve{\gamma}_h^{\mathrm{VB}})^2}{\gamma_h^2}-\left(1-\frac{(L+M)\sigma^2}{\gamma_h^2}\right)\frac{\breve{\gamma}_h^{\mathrm{VB}}}{\gamma_h}+\frac{LM\sigma^4}{\gamma_h^4}\right)}{\sigma^6}, \tag{132}$$

$$\frac{\partial \widehat{\gamma}_h}{\partial c_{a_h}^2c_{b_h}^2} = \frac{4\gamma_h^2\sigma^2}{4\gamma_hc_{a_h}^4c_{b_h}^4\sqrt{(M-L)^2+\frac{4\gamma_h^2}{c_{a_h}^2c_{b_h}^2}}}$$
$$= \frac{\sigma^4}{2\gamma_hc_{a_h}^4c_{b_h}^4\left(1-\left(\frac{\breve{\gamma}_h^{\mathrm{VB}}}{\gamma_h}+\frac{(M+L)\sigma^2}{2\gamma_h^2}\right)\right)}. \tag{133}$$

Here, we used Eqs.(21) and (81) to obtain Eqs.(130) and (131), and Eqs.(17) and (82) to obtain Eqs.(132) and (133), respectively. Eq.(85) is obtained by multiplying Eqs.(130) and (131), while Eq.(86) is obtained by multiplying Eqs.(132) and (133).

Taking the difference between the derivatives (85) and (86), and then using Eqs.(82) and (84), we have

$$\frac{\partial(F_h^{\mathrm{Posi}}-F_h^{\mathrm{Null}})}{\partial c_{a_h}^2c_{b_h}^2} = \frac{\partial F_h^{\mathrm{Posi}}}{\partial c_{a_h}^2c_{b_h}^2} - \frac{\partial F_h^{\mathrm{Null}}}{\partial c_{a_h}^2c_{b_h}^2}$$
$$= -\frac{1}{\sigma^2c_{a_h}^2c_{b_h}^2}\left(\gamma_h\left(\gamma_h-\widehat{\gamma}_h\right)-(\underline{\gamma}_h^{\mathrm{VB}})^2\right). \tag{134}$$

The following can be obtained from Eqs.(82) and (83), respectively:

$$\left(\gamma_h(\gamma_h-\breve{\gamma}_h^{\mathrm{VB}})-\frac{(L+M)\sigma^2}{2}\right)^2 = \frac{(L+M)^2\sigma^4}{4}-LM\sigma^4+\frac{\sigma^4}{c_{a_h}^2c_{b_h}^2}\gamma_h^2, \tag{135}$$

$$\left((\underline{\gamma}_h^{\mathrm{VB}})^2 - \frac{(L+M)\sigma^2}{2}\right)^2 = \frac{(L+M)^2\sigma^4}{4} - LM\sigma^4 + \frac{\sigma^4}{c_{a_h}^2 c_{b_h}^2}(\underline{\gamma}_h^{\mathrm{VB}})^2. \tag{136}$$

Eqs.(135) and (136) imply that

$$\gamma_h(\gamma_h - \breve{\gamma}_h^{\mathrm{VB}}) > (\underline{\gamma}_h^{\mathrm{VB}})^2 \qquad \text{when} \qquad \gamma_h > \underline{\gamma}_h^{\mathrm{VB}}.$$

Therefore, Eq.(134) is negative, which completes the proof of Lemma 21. ■

G.4 Proof of Lemma 26

Lemma 25 immediately leads to the EVB shrinkage estimator (26). We can find the value of $c_{a_h}c_{b_h}$ at the positive EVB local solution by combining the condition (82) for the VB estimator and the condition (92) for the EVB estimator:

$$\left(\gamma_h - \frac{\gamma_h\breve{\gamma}_h^{\mathrm{EVB}}}{\breve{\gamma}_h^{\mathrm{EVB}} + \frac{M\sigma^2}{\gamma_h}}\right)\left(\gamma_h - \frac{\gamma_h\breve{\gamma}_h^{\mathrm{EVB}}}{\breve{\gamma}_h^{\mathrm{EVB}} + \frac{L\sigma^2}{\gamma_h}}\right) = \frac{\sigma^4}{c_{a_h}^2 c_{b_h}^2}$$
$$\frac{LM\sigma^4}{\gamma_h\breve{\gamma}_h^{\mathrm{EVB}}} = \frac{\sigma^4}{c_{a_h}^2 c_{b_h}^2},$$

which gives the former equation in Eq.(94). Similarly, using Eqs.(19) and (92), we have

$$\widehat{\delta}_h = \frac{c_{a_h}^2}{\sigma^2}\left(\gamma_h - \frac{\gamma_h\breve{\gamma}_h^{\mathrm{EVB}}}{\breve{\gamma}_h^{\mathrm{EVB}} + \frac{M\sigma^2}{\gamma_h}}\right)$$
$$= \frac{c_{a_h}^2 M}{\gamma_h}\left(1 + \frac{L\sigma^2}{\gamma_h\breve{\gamma}_h^{\mathrm{EVB}}}\right).$$

Using the assumption that $c_{a_h} = c_{b_h}$ and therefore $c_{a_h}^2 = c_{a_h}c_{b_h}$, we obtain the latter equation in Eq.(94). The equations in Eq.(93) are simply obtained from Lemma 20.

Finally, applying Eq.(92) to the free energy (80), we have

$$F_h^{\mathrm{EVB-Posi}} = -M\log\left(1 - \frac{\gamma_h\breve{\gamma}_h^{\mathrm{EVB}}}{\gamma_h\breve{\gamma}_h^{\mathrm{EVB}} + M\sigma^2}\right) - L\log\left(1 - \frac{\gamma_h\breve{\gamma}_h^{\mathrm{EVB}}}{\gamma_h\breve{\gamma}_h^{\mathrm{EVB}} + L\sigma^2}\right) - \frac{\gamma_h\breve{\gamma}_h^{\mathrm{EVB}}}{\sigma^2},$$

which leads to Eq.(95). This completes the proof of Lemma 26. ■

G.5 Proof of Lemma 27

The derivative is

$$\frac{\partial\Phi}{\partial z} = \frac{1 - \frac{1}{z+1} - \log(z+1)}{z^2},$$

which is negative for $z > 0$ because

$$\frac{1}{z+1} + \log(z+1) > 1.$$

This completes the proof of Lemma 27. ■

G.6 Poof of Lemma 28

Since $\Phi(z)$ is decreasing, $\Xi(\tau;\alpha)$ is upper-bounded by

$$\Xi(\tau;\alpha) = \Phi(\tau) + \Phi\left(\frac{\tau}{\alpha}\right) \leq 2\Phi(\tau) = \Xi(\tau;1).$$

Therefore, the unique zero-cross point $\underline{\tau}$ of $\Xi(\tau;\alpha)$ is no greater than the unique zero-cross point $\underline{z}$ of $\Phi(z)$:

$$\underline{\tau} \leq \underline{z}.$$

For obtaining the lower-bound $\underline{\tau} > \sqrt{\alpha}$, it suffices to show that $\Xi(\sqrt{\alpha};\alpha) > 0$. Below, we prove that the following function is decreasing and positive for $0 < \alpha \leq 1$:

$$g(\alpha) \equiv \frac{\Xi(\sqrt{\alpha};\alpha)}{\sqrt{\alpha}}.$$

From the definition (23) of $\Xi(\tau;\alpha)$, we have

$$g(\alpha) = \left(1 + \frac{1}{\alpha}\right)\log(\sqrt{\alpha}+1) - \log\sqrt{\alpha} - \frac{1}{\sqrt{\alpha}}.$$

The derivative is given by

$$\begin{aligned}\frac{\partial g}{\partial\sqrt{\alpha}} &= \frac{\left(1+\frac{1}{\alpha}\right)}{\sqrt{\alpha}+1} - \frac{2}{\alpha^{3/2}}\log(\sqrt{\alpha}+1) - \frac{1}{\sqrt{\alpha}} + \frac{1}{\alpha} \\ &= -\frac{2}{\alpha^{3/2}}\left(\log(\sqrt{\alpha}+1) + \frac{1}{\sqrt{\alpha}+1} - 1\right) \\ &< 0,\end{aligned}$$

which implies that $g(\alpha)$ is decreasing. Since

$$g(1) = 2\log 2 - 1 \approx 0.3863 > 0,$$

$g(\alpha)$ is positive for $0 < \alpha \leq 1$, which completes the proof of Lemma 28. ■

G.7 Proof of Lemma 30

Since

$$\begin{aligned}\frac{\partial\psi_0}{\partial x} &= 1 - \frac{1}{x}, \\ \frac{\partial^2\psi_0}{\partial x^2} &= \frac{1}{x^2} > 0,\end{aligned} \tag{137}$$

$\psi_0(x)$ is differentiable and strictly convex for $x > 0$ with its minimizer at $x = 1$. $\psi_1(x)$ is continuous for $x \geq \underline{x}$, and Eq.(99) implies that $\psi_1(x_h) \propto F_h^{\mathrm{EVB-Posi}}$. Accordingly, $\psi_1(x) \leq 0$ for $x \geq \underline{x}$, where the equality holds when $x = \underline{x}$. This equality implies that $\psi(x)$ is continuous. Since $\underline{x} > 1$, $\psi(x)$ shares the same minimizer with $\psi_0(x)$ at $x = 1$ (see Figure 3).

Hereafter, we investigate $\psi_1(x)$ and $\psi(x)$ for $x \geq \underline{x}$. By differentiating Eqs.(43) and (36), respectively, we have

$$\frac{\partial \psi_1}{\partial \tau} = -\left(\frac{\frac{\tau^2}{\alpha} - 1}{(\tau+1)\left(\frac{\tau}{\alpha}+1\right)} \right) < 0, \tag{138}$$

$$\frac{\partial \tau}{\partial x} = \frac{1}{2}\left(1 + \frac{x-(1+\alpha)}{\sqrt{(x-(1+\alpha))^2 - 4\alpha}} \right) > 0. \tag{139}$$

Substituting

$$x = x(\tau;\alpha) = (1+\tau)\left(1+\frac{\alpha}{\tau}\right) = 1+\alpha+\tau+\alpha\tau^{-1} \tag{35}$$

into Eq.(139), we have

$$\frac{\partial \tau}{\partial x} = \frac{\tau^2}{\alpha\left(\frac{\tau^2}{\alpha}-1\right)}. \tag{140}$$

Multiplying Eqs.(138) and (140) gives

$$\frac{\partial \psi_1}{\partial x} = \frac{\partial \psi_1}{\partial \tau}\frac{\partial \tau}{\partial x} = -\left(\frac{\tau^2}{\alpha(\tau+1)\left(\frac{\tau}{\alpha}+1\right)} \right) = -\frac{\tau}{x} < 0, \tag{141}$$

which implies that $\psi_1(x)$ is decreasing for $x > \underline{x}$.

Let us focus on the thresholding point of $\psi(x)$ at $x = \underline{x}$. Eq.(141) does not converge to zero for $x \to \underline{x} + 0$ but stay negative. On the other hand, $\psi_0(x)$ is differentiable at $x = \underline{x}$. Consequently, $\psi(x)$ has a discontinuously decreasing derivative, i.e., $\lim_{x\to\underline{x}-0} \partial\psi/\partial x > \lim_{x\to\underline{x}+0} \partial\psi/\partial x$, at $x = \underline{x}$.

Finally, we prove the strict quasi-convexity of $\psi(x)$. Taking the sum of Eqs.(137) and (141) gives

$$\frac{\partial \psi}{\partial x} = \frac{\partial \psi_0}{\partial x} + \frac{\partial \psi_1}{\partial x} = 1 - \frac{1+\tau}{x} = 1 - \frac{1+\tau}{1+\tau+\alpha+\alpha\tau^{-1}} > 0.$$

This means that $\psi(x)$ is increasing for $x > \underline{x}$. Since $\psi_0(x)$ is strictly convex and increasing at $x = \underline{x}$, and $\psi(x)$ is continuous, $\psi(x)$ is strictly quasi-convex. This completes the proof of Lemma 30. ■

G.8 Proof of Lemma 31

The strict convexity of $\psi_0(x)$ and the strict quasi-convexity of $\psi(x)$ also hold for $\psi_0(\gamma_h^2\sigma^{-2}/M)$ and $\psi(\gamma_h^2\sigma^{-2}/M)$ as functions of σ^{-2} (for $\gamma_h > 0$). Because of the different scale factor γ_h^2/M for each $h = 1, \ldots, L$, each of $\psi_0(\gamma_h^2\sigma^{-2}/M)$ and $\psi(\gamma_h^2\sigma^{-2}/M)$ has a minimizer at a different position:

$$\sigma^{-2} = \frac{M}{\gamma_h^2}.$$

The strict quasi-convexity of ψ_0 and ψ guarantees that $\Omega(\sigma^{-2})$ is decreasing for

$$0 < \sigma^{-2} < \frac{M}{\gamma_1^2},$$

and increasing for

$$\frac{M}{\gamma_L^2} < \sigma^{-2} < \infty.$$

This proves Lemma 31. ■

G.9 Proof of Lemma 34

The derivative of Eq.(40) with respect to σ^{-2} is given by

$$\frac{\partial \Omega}{\partial \sigma^{-2}} = \frac{1}{L}\left(\sum_{h=1}^{H} \frac{\gamma_h^2}{M}\frac{\partial \psi}{\partial x} + \sum_{h=H+1}^{L} \frac{\gamma_h^2}{M}\frac{\partial \psi_0}{\partial x}\right). \tag{142}$$

By using Eqs.(137) and (141), Eq.(142) can be written as

$$\begin{aligned}
\frac{\partial \Omega}{\partial \sigma^{-2}} &= \frac{1}{L}\left(\sum_{h=1}^{L} \frac{\gamma_h^2}{M}\frac{\partial \psi_0}{\partial x} + \sum_{h=1}^{H} \theta\left(x_h \geq \underline{x}\right)\frac{\gamma_h^2}{M}\frac{\partial \psi_1}{\partial x}\right) \\
&= \frac{1}{L}\left(\sum_{h=1}^{L} \frac{\gamma_h^2}{M}\left(1 - \frac{1}{x_h}\right) - \sum_{h=1}^{H} \theta\left(x_h \geq \underline{x}\right)\frac{\gamma_h^2 \tau_h}{M x_h}\right) \\
&= \frac{\sum_{h=1}^{L}\gamma_h^2}{LM} - \sigma^2 - \frac{1}{L}\sum_{h=1}^{H} \theta\left(\tau_h \geq \underline{\tau}\right)\sigma^2 \tau_h.
\end{aligned} \tag{143}$$

Here, we also used the definition (33) of x_h. Using Eq.(34), Eq.(143) can be written as

$$\begin{aligned}
\frac{\partial \Omega}{\partial \sigma^{-2}} &= \frac{\sum_{h=1}^{L}\gamma_h^2}{LM} - \sigma^2 - \sum_{h=1}^{H} \theta\left(\gamma_h \geq \underline{\gamma}^{\mathrm{EVB}}\right)\frac{\gamma_h \breve{\gamma}_h^{\mathrm{EVB}}}{LM} \\
&= -\sigma^2 + \frac{\sum_{h=1}^{H}\gamma_h\left(\gamma_h - \widehat{\gamma}_h^{\mathrm{EVB}}\right) + \sum_{h=H+1}^{L}\gamma_h^2}{LM}.
\end{aligned}$$

Here, we also used the definition (24) of $\widehat{\gamma}_h^{\mathrm{EVB}}$. Using the definition (102) and Lemma 33, we can replace $\widehat{\gamma}_h^{\mathrm{EVB}}$ and H with $\breve{\gamma}_h^{\mathrm{EVB}}$ and $\widehat{H}$, respectively, which completes the proof of Lemma 34. ■

G.10 Proof of Lemma 36

By substituting the lower-bound in Eq.(104) into Eq.(101), we obtain

$$\Theta \geq -\sigma^2 + \frac{\widehat{H}(M+L)\sigma^2 + \sum_{h=\widehat{H}+1}^{L}\gamma_h^2}{LM}.$$

This implies that $\Theta > 0$ unless the following hold:

$$\widehat{H} < \frac{LM}{M+L} = \frac{L}{1+\alpha},$$
$$\sigma^2 \geq \frac{\sum_{h=\widehat{H}+1}^{L} \gamma_h^2}{LM - \widehat{H}(M+L)}.$$

Therefore, no local minimum exists if either of these conditions is violated. This completes the proof of Lemma 36. ■

G.11 Proof of Lemma 37

Focusing on the support

$$\log \underline{y} < t < \log \overline{y}$$

of the LSMP distribution (109), we define

$$\begin{aligned} f(t) \equiv 2 \log p^{\mathrm{LSMP}}(t) &= 2 \log \frac{\sqrt{(e^t - \underline{y})(\overline{y} - e^t)}}{2\pi\alpha} \\ &= \log(-e^{2t} + (\underline{y} + \overline{y})e^t - \underline{y}\overline{y}) + \text{const.}. \end{aligned}$$

Let

$$u(t) \equiv (e^t - \underline{y})(\overline{y} - e^t) = -e^{2t} + (\underline{y} + \overline{y})e^t - \underline{y}\overline{y} > 0,$$

and let

$$\begin{aligned} v(t) \equiv \frac{\partial u}{\partial t} &= -2e^{2t} + (\underline{y} + \overline{y})e^t = u - e^{2t} + \underline{y}\overline{y}, \\ w(t) \equiv \frac{\partial^2 u}{\partial t^2} &= -4e^{2t} + (\underline{y} + \overline{y})e^t = v - 2e^{2t}, \end{aligned}$$

be the first and the second derivatives of u.

Therefore, the first and the second derivatives of $f(t)$ are given by

$$\begin{aligned} \frac{\partial f}{\partial t} &= \frac{v}{u}, \\ \frac{\partial^2 f}{\partial t^2} &= \frac{uw - v^2}{u^2} \\ &= -\frac{e^t\left((\underline{y} + \overline{y})e^{2t} - 4\underline{y}\overline{y}e^t + (\underline{y} + \overline{y})\underline{y}\overline{y}\right)}{u^2} \\ &= -\frac{e^t(\underline{y} + \overline{y})}{u^2}\left(\left(e^t - \frac{2\underline{y}\overline{y}}{(\underline{y} + \overline{y})}\right)^2 + \frac{\underline{y}\overline{y}\left(\overline{y} - \underline{y}\right)^2}{(\underline{y} + \overline{y})^2}\right) \\ &\leq 0. \end{aligned}$$

This proves the log-concavity of the LSMP distribution $p^{\mathrm{LSMP}}(t)$, and completes the proof of Lemma 37. ■

G.12 Proof of Lemma 42

Lemma 41 states that $\Omega_0(\sigma^{-2})$, defined by Eq.(116), is quasi-convex with its minimizer at

$$\sigma^{-2} = \left(\frac{\sum_{h=H^*+1}^{L} \gamma_h^2}{(L-H^*)M} \right)^{-1} = \sigma^{*-2}.$$

Since $\Omega_1(\sigma^{-2})$, defined by Eq.(115), is a sum of strictly quasi-convex functions with their minimizers at $\sigma^{-2} = M/\gamma_h^2 < \sigma^{*-2}$ for $h = 1, \ldots, H^*$, our objective $\Omega(\sigma^{-2})$, given by Eq.(114), is non-decreasing (increasing if $H^* > 0$) for

$$\sigma^{-2} \geq \sigma^{*-2}.$$

Since Eq.(118) implies that $\underline{\sigma}_{H^*+1}^{-2} > \sigma^{*-2}$, $\Omega(\sigma^{-2})$ is non-decreasing (increasing if $\xi > 0$) for $\sigma^{-2} > \underline{\sigma}_{H^*+1}^{-2}$, which completes the proof of Lemma 42. ■

G.13 Proof of Lemma 43

Lemma 41 states that $\Omega_0(\sigma^{-2})$ is strictly convex in the range $0 < \sigma^{-2} < \underline{\sigma}_{H^*+1}^2$, and minimized at $\sigma^{-2} = \sigma^{*-2}$. Since Eq.(118) implies that $\sigma^{*-2} < \underline{\sigma}_{H^*+1}^2$, $\Omega_0(\sigma^{-2})$ is increasing at $\sigma^{-2} = \underline{\sigma}_{H^*+1}^2 - 0$. Since $\Omega_1(\sigma^{-2})$ is a sum of strictly quasi-convex functions with their minimizers at $\sigma^{-2} = M/\gamma_h^2 < \sigma^{*-2}$ for $h = 1, \ldots, H^*$, $\Omega(\sigma^{-2})$ is also increasing at $\sigma^{-2} = \underline{\sigma}_{H^*+1}^2 - 0$.

Let us investigate the sign of the derivative Θ of $\Omega(\sigma^{-2})$ at $\sigma^{-2} = \underline{\sigma}_{H^*}^2 + 0 \in \mathbb{B}_{H^*}$. Substituting the upper-bound in Eq.(104) into Eq.(101), we have

$$\begin{aligned} \Theta &< -\sigma^2 + \frac{H^*(\sqrt{M}+\sqrt{L})^2\sigma^2 + \sum_{h=H^*+1}^{L} \gamma_h^2}{LM} \\ &= -\sigma^2 + \frac{H^*(\sqrt{M}+\sqrt{L})^2\sigma^2 + (L-H^*)M\sigma^{*2}}{LM}. \end{aligned} \tag{144}$$

The right-hand side of Eq.(144) is negative if the following hold:

$$\xi = \frac{H^*}{L} < \frac{M}{(\sqrt{M}+\sqrt{L})^2} = \frac{1}{(1+\sqrt{\alpha})^2}, \tag{145}$$

$$\sigma^2 > \frac{(L-H^*)M\sigma^{*2}}{LM - H^*(\sqrt{M}+\sqrt{L})^2} = \frac{(1-\xi)\sigma^{*2}}{1-\xi(1+\sqrt{\alpha})^2}. \tag{146}$$

Assume that the first condition (145) holds. Then, the second condition (146) holds at $\sigma^{-2} = \underline{\sigma}_{H^*}^2 + 0$, if

$$\underline{\sigma}_{H^*}^{-2} < \frac{1-\xi(1+\sqrt{\alpha})^2}{(1-\xi)}\sigma^{*-2},$$

or equivalently,

$$y_{H^*} = \frac{\gamma_{H^*}^2}{M\sigma^{*2}} = \underline{x} \cdot \frac{\underline{\sigma}_{H^*}^2}{\sigma^{*2}} > \frac{\underline{x}(1-\xi)}{1-\xi(1+\sqrt{\alpha})^2},$$

which completes the proof of Lemma 43. ■

G.14 Proof of Lemma 44

In the range $0 < \sigma^{-2} < \underline{\sigma}^2_{H^*}$, the estimated rank (102) is bounded as

$$0 \leq \widehat{H} \leq H^* - 1.$$

Substituting the upper-bound in Eq.(104) into Eq.(101), we have

$$\begin{aligned}\Theta &< -\sigma^2 + \frac{\widehat{H}(\sqrt{M}+\sqrt{L})^2\sigma^2 + \sum_{h=\widehat{H}+1}^{H^*}\gamma_h^2 + \sum_{h=H^*+1}^{L}\gamma_h^2}{LM} \\ &= -\sigma^2 + \frac{\widehat{H}(\sqrt{M}+\sqrt{L})^2\sigma^2 + \sum_{h=\widehat{H}+1}^{H^*}\gamma_h^2 + (L-H^*)M\sigma^{*2}}{LM}. \end{aligned} \tag{147}$$

The right-hand side of Eq.(147) is negative, if the following hold:

$$\frac{\widehat{H}}{L} < \frac{M}{(\sqrt{M}+\sqrt{L})^2} = \frac{1}{(1+\sqrt{\alpha})^2}, \tag{148}$$

$$\sigma^2 > \frac{\sum_{h=\widehat{H}+1}^{H^*}\gamma_h^2 + (L-H^*)M\sigma^{*2}}{LM - \widehat{H}(\sqrt{M}+\sqrt{L})^2}. \tag{149}$$

Assume that

$$\xi = \frac{H^*}{L} < \frac{1}{(1+\sqrt{\alpha})^2}.$$

Then, both of the conditions (148) and (149) hold anywhere in $0 < \sigma^{-2} < \underline{\sigma}^2_{H^*}$, if the following holds

$$\underline{\sigma}^{-2}_{\widehat{H}+1} < \frac{LM - \widehat{H}(\sqrt{M}+\sqrt{L})^2}{\sum_{h=\widehat{H}+1}^{H^*}\gamma_h^2 + (L-H^*)M\sigma^{*2}} \qquad \text{for} \qquad \widehat{H} = 0, \ldots, H^*-1. \tag{150}$$

Since the sum $\sum_{h=\widehat{H}+1}^{H^*}\gamma_h^2$ in the right-hand side of Eq.(150) is upper-bounded as

$$\sum_{h=\widehat{H}+1}^{H^*}\gamma_h^2 \leq (H^* - \widehat{H})\gamma^2_{\widehat{H}+1},$$

Eq.(150) holds if

$$\begin{aligned}\underline{\sigma}^{-2}_{\widehat{H}+1} &< \frac{LM - \widehat{H}(\sqrt{M}+\sqrt{L})^2}{(H^*-\widehat{H})\gamma^2_{\widehat{H}+1} + (L-H^*)M\sigma^{*2}} \\ &= \frac{1 - \frac{\widehat{H}}{L}(1+\sqrt{\alpha})^2}{(\xi - \frac{\widehat{H}}{L})\frac{\gamma^2_{\widehat{H}+1}}{M} + (1-\xi)\sigma^{*2}} \qquad \text{for} \qquad \widehat{H} = 0, \ldots, H^*-1. \end{aligned} \tag{151}$$

Using Eq.(100), the condition (151) is rewritten as

$$\frac{\gamma^2_{\widehat{H}+1}}{M\underline{x}} > \frac{(\xi - \frac{\widehat{H}}{L})\frac{\gamma^2_{\widehat{H}+1}}{M} + (1-\xi)\sigma^{*2}}{1 - \frac{\widehat{H}}{L}(1+\sqrt{\alpha})^2}$$

$$\left(1-\frac{\widehat{H}}{L}(1+\sqrt{\alpha})^2\right)\frac{\gamma_{\widehat{H}+1}^2}{M\sigma^{*2}} > (\xi\underline{x}-\frac{\widehat{H}}{L}\underline{x})\frac{\gamma_{\widehat{H}+1}^2}{M\sigma^{*2}}+(1-\xi)\underline{x},$$

or equivalently

$$y_{\widehat{H}+1}=\frac{\gamma_{\widehat{H}+1}^2}{M\sigma^{*2}} > \frac{(1-\xi)\,\underline{x}}{\left(1-\xi\underline{x}+\frac{\widehat{H}}{L}\,(\underline{x}-(1+\sqrt{\alpha})^2)\right)} \qquad \text{for} \qquad \widehat{H}=0,\ldots,H^*-1. \quad (152)$$

Note that $\underline{x} > \overline{y} = (1+\sqrt{\alpha})^2$. Further bounding both sides, we have the following sufficient condition for Eq.(152) to hold:

$$y_{H^*} > \frac{(1-\xi)\underline{x}}{\max\,(0,1-\xi\underline{x})}.$$

Thus, we obtain the conditions (121) and (122) for Θ to be negative anywhere in $0<\sigma^{-2}<\underline{\sigma}_{H^*}^2$, which completes the proof of Lemma 44. ■

Appendix H. Detailed Description of Overlap Method

The overlap (OL) method (Hoyle, 2008) minimizes the following approximation to the negative log of the marginal likelihood (58) over the hypothetical model rank $H=1,\ldots,L$:[7]

$$\begin{aligned}
2F^{\mathrm{OL}} &\approx -2\log p(\boldsymbol{V}) \\
&= (LM-H(L-H-2))\log(2\pi)+L\log\pi-2\sum_{h=1}^{H}\log\left(\frac{\Gamma\left((M-h+1)/2\right)}{\Gamma\left((M-L-h+1)/2\right)}\right) \\
&\quad + H(M-L)\left(1-\log\left(M-L\right)\right)+\sum_{h=1}^{H}\sum_{l=H+1}^{L}\log\left(\gamma_h^2-\gamma_l^2\right)+(M-L)\sum_{h=1}^{H}\log\gamma_h^2 \\
&\quad + (M-H)\sum_{h=1}^{H}\log\left(\frac{1}{\widehat{\sigma}^{2\ \mathrm{OL}}}-\frac{1}{\widehat{\lambda}_h^{\mathrm{OL}}}\right)-\sum_{h=1}^{H}\left(\frac{1}{\widehat{\sigma}^{2\ \mathrm{OL}}}-\frac{1}{\widehat{\lambda}_h^{\mathrm{OL}}}\right)\gamma_h^2 \\
&\quad + (L+2)\left(\sum_{h=1}^{H}\log\widehat{\lambda}_h^{\mathrm{OL}}+(M-H)\log\widehat{\sigma}^{2\ \mathrm{OL}}\right)+\sum_{l=1}^{L}\frac{\gamma_l^2}{\widehat{\sigma}^{2\ \mathrm{OL}}},
\end{aligned}$$

where $\Gamma(\cdot)$ denotes the Gamma function, and $\{\widehat{\lambda}_h^{\mathrm{OL}}\}$ and $\widehat{\sigma}^{2\ \mathrm{OL}}$ are estimators for $\lambda_h = b_h^2+\sigma^2$ and σ^2, computed by iterating the following equations until convergence:

$$\begin{aligned}
\widehat{\lambda}_h^{\mathrm{OL}} = \frac{\gamma_h^2}{2(L+2)}\Bigg(1-\frac{(M-H-(L+2))\widehat{\sigma}^{2\ \mathrm{OL}}}{\gamma_h^2} \\
+\sqrt{\left(1-\frac{(M-H-(L+2))\widehat{\sigma}^{2\ \mathrm{OL}}}{\gamma_h^2}\right)^2-\frac{4(L+2)\widehat{\sigma}^{2\ \mathrm{OL}}}{\gamma_h^2}}\Bigg),
\end{aligned} \quad (153)$$

[7] Our description is slightly different from Hoyle (2008), because our model (1) does not have the mean parameter shared over the samples.

$$\widehat{\sigma}^{2\,\mathrm{OL}} = \frac{1}{(M-H)}\left(\sum_{l=1}^{L}\frac{\gamma_l^2}{L} - \sum_{h=1}^{H}\widehat{\lambda}_h^{\mathrm{OL}}\right). \tag{154}$$

When iterating Eqs.(153) and (154), $\widehat{\lambda}_h^{\mathrm{OL}}$ can be a complex number. In such a case, the hypothetical H is rejected. Otherwise, F^{OL} is evaluated after convergence, and $\widehat{H}^{\mathrm{OL}}$ that minimizes F^{OL} is chosen.

For the null hypothesis, the negative log likelihood is given by

$$2F^{\mathrm{OL}} = -2\log P(\boldsymbol{V}) = LM\left(\log\left(\frac{2\pi}{LM}\sum_{l=1}^{L}\gamma_l^2\right)+1\right) \qquad \text{for} \qquad H=0.$$

References

H. Attias. Inferring parameters and structure of latent variable models by variational Bayes. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence (UAI-99)*, pages 21–30, San Francisco, CA, 1999. Morgan Kaufmann.

Z. Bai and J. W. Silverstein. *Spectral Analysis of Large Dimensional Random Matrices*. Springer, 2010.

J. Baik and J. W. Silverstein. Eigenvalues of large sample covariance matrices of spiked population models. *Journal of Multivariate Analysis*, 97(6):1382–1408, 2006.

C. M. Bishop. Variational principal components. In *Proceedings of International Conference on Artificial Neural Networks*, volume 1, pages 514–509, 1999a.

C. M. Bishop. Bayesian principal components. In *Advances in Neural Information Processing Systems*, volume 11, pages 382–388, 1999b.

C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, New York, NY, USA, 2006.

C. M. Bishop and M. E. Tipping. Variational relevance vector machines. In *Proceedings of the Sixteenth Conference Annual Conference on Uncertainty in Artificial Intelligence*, pages 46–53, 2000.

D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

J. P. Bouchaud and M. Potters. *Theory of Financial Risk and Derivative Pricing—From Statistical Physics to Risk Management, Second Edition*. Cambridge University Press, 2003.

E. J. Candès and T. Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE Transactions Information Theory*, 52(12):5406–5425, 2006.

S. Dharmadhikari and K. Joag-dev. *Unimodality, Convexity, and Applications*. Academic Press, 1988.

Z. Ghahramani and M. J. Beal. Graphical models and variational methods. In *Advanced Mean Field Methods*, pages 161–177. MIT Press, 2001.

D. C. Hoyle. Automatic PCA dimension selection for high dimensional data and small sample sizes. *Journal of Machine Learning Research*, 9:2733–2759, 2008.

D. C. Hoyle and M. Rattray. Principal-component-analysis eigenvalue spectra from data with symmetry-breaking structure. *Physical Review E*, 69(026124), 2004.

I. A. Ibragimov. On the composition of unimodal distributions. *Theory of Probability and Its Applications*, 1(2):255–260, 1956.

A. Ilin and T. Raiko. Practical approaches to principal component analysis in the presence of missing values. *Journal of Machine Learning Research*, 11:1957–2000, 2010.

T. S. Jaakkola and M. I. Jordan. Bayesian parameter estimation via variational methods. *Statistics and Computing*, 10:25–37, 2000.

I. M. Johnstone. On the distribution of the largest eigenvalue in principal components analysis. *Annals of Statistics*, 29:295–327, 2001.

Y. J. Lim and Y. W. Teh. Variational Bayesian approach to movie rating prediction. In *Proceedings of KDD Cup and Workshop*, 2007.

D. J. C. Mackay. Local minima, symmetry-breaking, and model pruning in variational free energy minimization, 2001. URL `http://www.inference.phy.cam.ac.uk/mackay/minima.pdf`.

V. A. Marčenko and L. A. Pastur. Distribution of eigenvalues for some sets of random matrices. *Mathematics of the USSR-Sbornik*, 1(4):457–483, 1967.

M. L. Mehta. *Random Matrices, Third Edition*. Academic Press, 2000.

T. P. Minka. Automatic choice of dimensionality for PCA. In *Advances in Neural Information Processing Systems*, volume 13, pages 598–604. MIT Press, 2001.

S. Nakajima and M. Sugiyama. Theoretical analysis of Bayesian matrix factorization. *Journal of Machine Learning Research*, 12:2579–2644, 2011.

S. Nakajima and M. Sugiyama. Analysis of empirical MAP and empirical partially Bayes: Can they be alternatives to variational Bayes? In *Proceedings of International Conference on Artificial Intelligence and Statistics*, volume 33, pages 20–28, 2014.

S. Nakajima, M. Sugiyama, and S. D. Babacan. On Bayesian PCA: Automatic dimensionality selection and analytic solution. In *Proceedings of 28th International Conference on Machine Learning (ICML2011)*, pages 497–504, Bellevue, WA, USA, Jun. 28–Jul.2 2011.

S. Nakajima, R. Tomioka, M. Sugiyama, and S. D. Babacan. Perfect dimensionality recovery by variational Bayesian PCA. In P. Bartlett, F. C. N. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 980–988, 2012.

S. Nakajima, M. Sugiyama, and S. D. Babacan. Variational Bayesian sparse additive matrix factorization. *Machine Learning*, 92:319–1347, 2013a.

S. Nakajima, M. Sugiyama, S. D. Babacan, and R. Tomioka. Global analytic solution of fully-observed variational Bayesian matrix factorization. *Journal of Machine Learning Research*, 14:1–37, 2013b.

B. Recht, M. Fazel, and P. A. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Reveiw*, 52(3):471–501, 2010.

S. Roweis and Z. Ghahramani. A unifying review of linear Gaussian models. *Neural Computation*, 11:305–345, 1999.

R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In J. C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 1257–1264, Cambridge, MA, 2008. MIT Press.

M. Sato, T. Yoshioka, S. Kajihara, K. Toyama, N. Goda, K. Doya, and M. Kawato. Hierarchical Bayesian estimation for MEG inverse problem. *Neuro Image*, 23:806–826, 2004.

M. Seeger. Sparse linear models: Variational approximate inference and Bayesian experimental design. In *Journal of Physics: Conference Series*, volume 197, 2009.

M. Seeger and G. Bouchard. Fast variational Bayesian inference for non-conjugate matrix factorization models. In *Proceedings of International Conference on Artificial Intelligence and Statistics*, La Palma, Spain, 2012.

M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society*, 61:611–622, 1999.

A. M. Tulino and S. Verdu. *Random Matrix Theory and Wireless Communications*. Now Publishers, 2004.

K. W. Wachter. The strong limits of random matrix spectra for sample matrices of independent elements. *Annals of Probability*, 6:1–18, 1978.

E. P. Wigner. On the distribution of the roots of certain symmetric matrices. *Annals of Mathematics*, 67(2):325–327, 1957.

Journal of Machine Learning Research 16 (2015) 3813-3847 Submitted 11/13; Revised 1/15; Published 12/15

Graphical Models via Univariate Exponential Family Distributions

Eunho Yang EUNHYANG@US.IBM.COM
IBM T.J. Watson Research Center
Yorktown Heights, NY 10598, USA

Pradeep Ravikumar PRADEEPR@CS.UTEXAS.EDU
Department of Computer Science
University of Texas, Austin
Austin, TX 78712, USA

Genevera I. Allen GALLEN@RICE.EDU
Department of Statistics
Rice University
Houston, TX 77005, USA

Zhandong Liu ZHANDONL@BCM.EDU
Department of Pediatrics-Neurology
Baylor College of Medicine
Houston, TX 77030, USA

Editor: Tommi Jaakkola

Abstract

Undirected graphical models, or Markov networks, are a popular class of statistical models, used in a wide variety of applications. Popular instances of this class include Gaussian graphical models and Ising models. In many settings, however, it might not be clear which subclass of graphical models to use, particularly for non-Gaussian and non-categorical data. In this paper, we consider a general sub-class of graphical models where the node-wise conditional distributions arise from exponential families. This allows us to derive *multivariate* graphical model distributions from *univariate* exponential family distributions, such as the Poisson, negative binomial, and exponential distributions. Our key contributions include a class of M-estimators to fit these graphical model distributions; and rigorous statistical analysis showing that these M-estimators recover the true graphical model structure exactly, with high probability. We provide examples of genomic and proteomic networks learned via instances of our class of graphical models derived from Poisson and exponential distributions.

Keywords: graphical models, model selection, sparse estimation

1. Introduction

Undirected graphical models, also known as Markov random fields, are an important class of statistical models that have been extensively used in a wide variety of domains, including statistical physics, natural language processing, image analysis, and medicine. The key idea in this class of models is to represent the joint distribution as a product of *clique-wise compatibility functions*. Given an underlying graph, each of these compatibility functions

depends only on a subset of variables within any clique of the underlying graph. Popular instances of such graphical models include Ising and Potts models (see references in Wainwright and Jordan (2008) for a varied set of applications in computer vision, text analytics, and other areas with discrete variables), as well as Gaussian Markov Random Fields (GMRFs), which are popular in many scientific settings for modeling real-valued data. A key modeling question that arises, however, is: how do we pick the clique-wise compatibility functions, or alternatively, how do we pick the form or sub-class of the graphical model distribution (e.g. Ising or Gaussian MRF)? For the case of discrete random variables, Ising and Potts models are popular choices; but these are not best suited for count-valued variables, where the values taken by any variable could range over the entire set of positive integers. Similarly, in the case of continuous variables, Gaussian Markov Random Fields (GMRFs) are a popular choice; but the distributional assumptions imposed by GMRFs are quite stringent. The marginal distribution of any variable would have to be Gaussian for instance, which might not hold in instances when the random variables characterizing the data are skewed (Liu et al., 2009). More generally, Gaussian random variables have thin tails, which might not capture fat-tailed events and variables. For instance, in the finance domain, the lack of modeling of fat-tailed events and probabilities has been suggested as one of the causes of the 2008 financial crisis (Acemoglu, 2009).

To address this modeling question, some have recently proposed non-parametric extensions of graphical models. Some, such as the non-paranormal (Liu et al., 2009; Lafferty et al., 2012) and copula-based methods (Dobra and Lenkoski, 2011; Liu et al., 2012a), use or learn transforms that Gaussianize the data, and then fit Gaussian MRFs to estimate network structure. Others, use non-parametric approximations, such as rank-based estimators, to the correlation matrix, and then fit a Gaussian MRF (Xue and Zou, 2012; Liu et al., 2012b). More broadly, there could be non-parametric methods that either learn the sufficient statistics functions, or learn transformations of the variables, and then fit standard MRFs over the transformed variables. However, the sample complexity of such classes of non-parametric methods is typically inferior to those that learn parametric models. Alternatively, and specifically for the case of multivariate count data, Lauritzen (1996); Bishop et al. (2007) have suggested combinatorial approaches to fitting graphical models, mostly in the context of contingency tables. These approaches, however, are computationally intractable for even moderate numbers of variables.

Interestingly, for the case of *univariate* data, we have a good understanding of appropriate statistical models to use. In particular, a count-valued random variable can be modeled using a Poisson distribution; call-times, time spent on websites, diffusion processes, and life-cycles can be modeled with an exponential distribution; other skewed variables can be modeled with gamma or chi-squared distributions. Here, we ask if we can extend this modeling toolkit from univariate distributions to multivariate graphical model distributions? Interestingly, recent state of the art methods for learning Ising and Gaussian MRFs (Meinshausen and Bühlmann, 2006; Ravikumar et al., 2010; Jalali et al., 2011) suggest a natural procedure deriving such multivariate graphical models from univariate distributions. The key idea in these recent methods is to learn the MRF graph structure by estimating node-neighborhoods, or by fitting node-conditional distributions of each node conditioned on the rest of the nodes. Indeed, these node-wise fitting methods have been shown to have strong computational as well as statistical guarantees. Here, we consider the general class of mod-

els obtained by the following construction: suppose the node-conditional distributions of each node conditioned on the rest of the nodes follows a univariate exponential family. By the Hammersley-Clifford Theorem (Lauritzen, 1996), and some algebra as derived in Besag (1974), these node-conditional distributions entail a global multivariate distribution that (a) factors according to cliques defined by the graph obtained from the node-neighborhoods, and (b) has a particular set of compatibility functions specified by the univariate exponential family. The resulting class of MRFs, which we call exponential family MRFs, broadens the class of models available off the shelf, from the standard Ising, indicator-discrete, and Gaussian MRFs.

Thus the class of exponential family MRFs provides a principled approach to model multivariate distributions and network structures among a large number of variables; by providing a natural way to "extend" univariate exponential families of distributions to the multivariate case, in many cases where multivariate extensions did not exist in an analytical or computationally tractable form. Potential applications for these exponential family graphical models abound. Networks of call-times, time spent on websites, diffusion processes, and life-cycles can be modeled with exponential graphical models; other skewed multivariate data can be modeled with gamma or chi-squared graphical models; while multivariate count data such as from website visits, user-ratings, crime and disease incident reports, and bibliometrics could be modeled via Poisson graphical models. A key motivating application for our research is multivariate count data from next-generation genomic sequencing technologies (Mortazavi et al., 2008). This technology produces read counts of the number of short RNA fragments that have been mapped back to a particular gene; and measures gene expression with less technical variation than, and is thus rapidly replacing, microarrays (Marioni et al., 2008). Univariate count data is typically modeled using Poisson or negative binomial distributions (Li et al., 2011). As Gaussian graphical models have been traditionally used to understand genomic relationships and estimate regulatory pathways from microarray data, Poisson and negative-binomial graphical models could thus be used to analyze this next-generation sequencing data. Furthermore, there is a proliferation of new technologies to measure high-throughput genomic variation in which the data is not even approximately Gaussian (single nucleotide polymorphisms, copy number, methylation, and micro-RNA and gene expression via next-generation sequencing). For this data, a more general class of high-dimensional graphical models could thus lead to important breakthroughs in understanding genomic relationships and disease networks.

The construction of the class of exponential family graphical models also suggests a natural method for fitting such models: node-wise neighborhood estimation via sparsity constrained node-conditional likelihood maximization. A main contribution of this paper is to provide a sparsistency analysis (or analysis of variable selection consistency) for the recovery of the underlying graph structure of this broad class of MRFs. We note that the presence of non-linearities arising from the generalized linear models (GLM) posed subtle technical issues not present in the linear case (Meinshausen and Bühlmann, 2006). Indeed, for the specific cases of logistic, and multinomial respectively, Ravikumar et al. (2010); Jalali et al. (2011) derive such a sparsistency analysis via fairly extensive arguments, but which were tuned to the specific cases; for instance they used the fact that the variables were bounded, and the specific structure of the corresponding GLMs. Here we generalize their analysis to general GLMs, which required a subtler analysis as well as a slightly modified

M-estimator. We note that this analysis might be of independent interest even outside the context of modeling and recovering graphical models. In recent years, there has been a trend towards unified statistical analyses that provide statistical guarantees for broad classes of models via general theorems (Negahban et al., 2012). Our result is in this vein and provides structure recovery for the class of sparsity constrained generalized linear models. We hope that the techniques we introduce might be of use to address the outstanding question of sparsity constrained M-estimation in its full generality.

There has been related work on the simple idea above of constructing joint distributions via specifying node-conditional distributions. Varin and Vidoni (2005); Varin et al. (2011) propose the class of composite likelihood models where the joint distribution is a function of the conditional distributions of subsets of nodes conditioned on other subsets. Besag (1974) discuss such joint distribution constructions in the context of node-conditional distributions belonging to exponential families, but for special cases of joint distributions such as pairwise models. In this paper, we consider the general case of higher-order graphical models for the joint distributions, and univariate exponential families for the node-conditional distributions. Moreover, a key contribution of the paper is that we provide tractable M-estimators with corresponding high-dimensional statistical guarantees and analysis for learning this class of graphical models even under high-dimensional statistical regimes.

Additionally, we note that a preliminary abridged version of this paper appeared at (Yang et al., 2012). In this manuscript, we provide a more in depth theoretical analysis along with several novel developments. Particularly, we provide a novel analytic framework on the sparsistency of our M-estimators that provide tighter finite-sample bounds, simpler proofs, and less restrictive assumptions than that of (Yang et al., 2012); these innovations are discussed further in Section 3.2. Further, we highlight and study several instances of our framework, relating our work to the existing literature on Gaussian MRFs and Ising models, as well as introducing two novel instances, the Poisson MRF and Exponential MRF. For each of these cases, we provide specific corollaries on conditions necessary for sparsistent recovery of the underlying graph structure. Finally, we also provide a greatly expanded experimental analysis of our class of MRFs and their M-estimators compared to that of (Yang et al., 2012). Focusing on two novel instances of our model, the Poisson and Exponential MRF, we study the theoretical rates, graph structural recovery, and robustness of our estimators through simulated examples. Further, we provide an additional case study on protein signaling networks using the Exponential MRF in Section 4.2.2.

2. Exponential Family Graphical Models

Suppose $X = (X_1, \ldots, X_p)$ is a random vector, with each variable X_i taking values in a set $\mathcal{X}$. Let $G = (V, E)$ be an undirected graph over the set of nodes $V := \{1, \ldots, p\}$ corresponding to the p variables $\{X_r\}_{r=1}^p$. The graphical model over X corresponding to G is a set of distributions that satisfy *Markov independence assumptions* with respect to the graph G (Lauritzen, 1996). By the Hammersley-Clifford theorem (Clifford, 1990), any such distribution that is strictly positive over its domain also factors according to the graph in the following way. Let $\mathcal{C}$ be a set of cliques (fully-connected subgraphs) of the graph G, and let $\{\phi_c(X_c)\}_{c \in \mathcal{C}}$ be a set of clique-wise sufficient statistics. With this notation, any strictly positive distribution of X within the graphical model family represented by the graph G

takes the form:

$$P(X) \quad \propto \quad \exp\Big\{ \sum_{c \in \mathcal{C}} \theta_c \phi_c(X_c) \Big\} \tag{1}$$

where $\{\theta_c\}$ are weights over the sufficient statistics. An important special case is a *pairwise graphical model*, where the set of cliques $\mathcal{C}$ consists of the set of nodes V and the set of edges E, so that

$$P(X) \quad \propto \quad \exp\Big\{ \sum_{r \in V} \theta_r \phi_r(X_r) + \sum_{(r,t) \in E} \theta_{rt} \phi_{rt}(X_r, X_t) \Big\}. \tag{2}$$

As previously discussed, an important question is how to select the form of the graphical model distribution, which under the above parametrization in (1), translates to the question of selecting the class of sufficient statistics, ϕ. As discussed in the introduction, it is of particular interest to derive such a graphical model distribution as a *multivariate* extension of specified *univariate* parametric distributions such as negative binomial, Poisson, and others. We next outline a subclass of graphical models that answer these questions via the simple construction: set the node-conditional distributions of each node conditioned on the rest of the nodes as following a univariate exponential family, and then derive the joint distribution that is consistent with these node-conditional distributions. Then, in Section 3, we will study how to learn the underlying graph structure, or the edge set E, for this general class of "exponential family" graphical models. We provide a natural sparsity-encouraging M-estimator, and sufficient conditions under which the M-estimator recovers the graph structure with high probability.

2.1 The Form of Exponential Family Graphical Models

A popular class of univariate distributions is the exponential family, whose distribution for a random variable Z is given by

$$P(Z) = \exp\Big\{ \theta\, B(Z) + C(Z) - D(\theta) \Big\}, \tag{3}$$

with sufficient statistics $B(Z)$, base measure $C(Z)$, and log-normalization constant $D(\theta)$. Such exponential family distributions include a wide variety of commonly used distributions such as Gaussian, Bernoulli, multinomial, Poisson, exponential, gamma, chi-squared, beta, and many others; any of which can be instantiated with particular choices of the functions $B(\cdot)$, and $C(\cdot)$. Such exponential family distributions are thus used to model a wide variety of data types including skewed continuous data and count data. Here, we ask if we can leverage this ability to model univariate data to also model the multivariate case. Let $X = (X_1, X_2, \ldots, X_p)$ be a p-dimensional random vector; and let $G = (V, E)$ be an undirected graph over p nodes corresponding to the p variables. Could we then derive a graphical model distribution over X with underlying graph G, from a particular choice of *univariate* exponential family distribution (3) above?

Consider the following construction. Set the distribution of X_r given the rest of nodes $X_{V \setminus r}$ to be given by the above univariate exponential family distribution (3), and where the canonical exponential family parameter θ is set to a linear combination of k-th order

products of univariate functions $\{B(X_t)\}_{t\in N(r)}$, where $N(r)$ is the set of neighbors of node r according to graph G. This gives the following conditional distribution:

$$P(X_r|X_{V\setminus r}) = \exp\Big\{B(X_r)\Big(\theta_r + \sum_{t\in N(r)} \theta_{rt}\, B(X_t) + \sum_{t_2,t_3\in N(r)} \theta_{r\,t_2t_3}\, B(X_{t_2})B(X_{t_3}) + \ldots + \sum_{t_2,\ldots,t_k\in N(r)} \theta_{r\,t_2\ldots t_k} \prod_{j=2}^{k} B(X_{t_j})\Big) + C(X_r) - \bar{D}(X_{V\setminus r})\Big\}, \tag{4}$$

where $C(X_r)$ is specified by the exponential family, and $\bar{D}(X_{V\setminus r})$ is the log-normalization constant. Notice that we use the notation $\bar{D}(\cdot)$ in case when we express the log-partition function in terms of random variables. That is, $\bar{D}(X_{V\setminus r}) := D\big(\theta(X_{V\setminus r})\big)$ where $\theta(X_{V\setminus r})$ is the canonical parameter θ derived from $X_{V\setminus r}$.

By the Hammersley-Clifford theorem, and some elementary calculation, this conditional distribution can be shown to specify the following unique joint distribution $P(X_1,\ldots,X_p)$:

Proposition 1 *Suppose $X = (X_1, X_2, \ldots, X_p)$ is a p-dimensional random vector, and its node-conditional distributions are specified by (4) given an undirected graph G. Then its joint distribution $P(X_1,\ldots,X_p)$ belongs to the graphical model represented by G, and is given by*

$$P(X) = \exp\Big\{\sum_{r\in V}\theta_r B(X_r) + \sum_{r\in V}\sum_{t\in N(r)} \theta_{rt}\, B(X_r)B(X_t) + \ldots + \sum_{r\in V}\sum_{t_2,\ldots,t_k\in N(r)} \theta_{r\ldots t_k}\, B(X_r)\prod_{j=2}^{k} B(X_{t_j}) + \sum_{r\in V} C(X_r) - A(\theta)\Big\} \tag{5}$$

where $A(\theta)$ is the log-normalization constant.

Note that the function $D(\cdot)$ (and hence $\bar{D}(\cdot)$) in (4) is the log-partition function of the node-conditional distribution, while the function $A(\cdot)$ in (5) in turn is the log-partition function of the joint distribution. Proposition 1, thus, provides an answer to our earlier question on selecting the form of a graphical model distribution given a univariate exponential family distribution. When the node-conditional distributions follow a univariate exponential family as in (4), there exists a unique graphical model distribution as specified by (5). One question that remains, however, is whether the above construction, beginning with (4), is the most general possible. In particular, note that the canonical parameter of the node-conditional distribution in (4) is a *tensor factorization* of the univariate sufficient statistic, which seems a bit stringent. Interestingly, by extending the argument from (Besag, 1974), which considers the special pairwise case, and the Hammersley-Clifford Theorem, we can show that indeed (4) and (5) have the most general form.

Theorem 2 *Suppose $X = (X_1, X_2, \ldots, X_p)$ is a p-dimensional random vector, and its node-conditional distributions are specified by an exponential family,*

$$P(X_r|X_{V\setminus r}) = \exp\Big\{E(X_{V\setminus r})\, B(X_r) + C(X_r) - \bar{D}(X_{V\setminus r})\Big\}, \tag{6}$$

where the function $E(X_{V\setminus r})$, the canonical parameter of exponential family, depends on the rest of all random variables except X_r (and hence the log-normalization constant $\bar{D}(X_{V\setminus r})$). Further, suppose the corresponding joint distribution factors according to the graph G, with the factors over cliques of size at most k. Then, the conditional distribution in (6) *necessarily has the tensor-factorized form in* (4)*, and the corresponding joint distribution has the form in* (5)*.*

Theorem 2 thus tells us that under the general assumptions that:

(a) the joint distribution is a graphical model that factors according to a graph G, and has clique-factors of size at most k, and

(b) its node-conditional distribution follows an exponential family,

it *necessarily* follows that the conditional and joint distributions are given by (4) and (5) respectively.

An important special case is when the joint graphical model distribution has clique factors of size at most two. From Theorem 2, the conditional distribution is given by

$$P(X_r|X_{V\setminus r}) = \exp\left\{\theta_r\, B(X_r) + \sum_{t\in N(r)} \theta_{rt}\, B(X_r)B(X_t) + C(X_r) - \bar{D}(X_{V\setminus r})\right\}, \tag{7}$$

while the joint distribution is given as

$$P(X) = \exp\left\{\sum_{r\in V} \theta_r B(X_r) + \sum_{(r,t)\in E} \theta_{rt}\, B(X_r)B(X_t) + \sum_{r\in V} C(X_r) - A(\theta)\right\}. \tag{8}$$

For many classes of models (e.g. general Ising, discrete CRFs), the log-partition function of the joint distribution, $A(\cdot)$, has no analytical form, and might even be intractable to compute, while the function $D(\cdot)$ typically is more amenable, and available in analytical form, since it is the log-partition function of a *univariate* exponential family distribution.

When the univariate sufficient statistic function $B(\cdot)$ is a linear function $B(X_r) = X_r$, then the conditional distribution in (7) is precisely a generalized linear model (McCullagh and Nelder, 1989) in canonical form,

$$P(X_r|X_{V\setminus r}) = \exp\left\{\theta_r\, X_r + \sum_{t\in N(r)} \theta_{rt}\, X_r\, X_t + C(X_r) - \bar{D}(X_{V\setminus r};\theta)\right\}, \tag{9}$$

where the canonical parameter of GLMs becomes $\theta_r + \sum_{t\in N(r)} \theta_{rt}\, X_t$. At the same time, the joint distribution has the form:

$$P(X) = \exp\left\{\sum_{r\in V} \theta_r X_r + \sum_{(r,t)\in E} \theta_{rt}\, X_r\, X_t + \sum_{r\in V} C(X_r) - A(\theta)\right\} \tag{10}$$

where the log-partition function $A(\cdot)$ in this case is defined as

$$A(\theta) := \log \int_X \exp\left\{\sum_{r\in V} \theta_r X_r + \sum_{(r,t)\in E} \theta_{rt}\, X_r\, X_t + \sum_{r\in V} C(X_r)\right\} dX. \tag{11}$$

We will now provide some examples of our general class of "exponential family" graphical model distributions, focusing on the case in (10) with linear functions $B(X_r) = X_r$. For each of these examples, we will also detail the domain, $\Theta := \{\theta : A(\theta) < +\infty\}$, of valid parameters that ensure that the density is normalizable. Indeed, such constraints on valid parameters are typically necessary for the distributions over countable discrete or continuous valued variables.

2.2 Gaussian Graphical Models

The popular Gaussian graphical model (Speed and Kiiveri, 1986) can be derived as an instance of the construction in Theorem 2, with the univariate Gaussian distribution as the exponential family distribution. The univariate Gaussian distribution with known variance σ^2 is given by

$$P(Z) \propto \exp\left\{\frac{\mu}{\sigma}\frac{Z}{\sigma} - \frac{Z^2}{2\sigma^2}\right\}$$

where $Z \in \mathbb{R}$, so that it can be seen to be an exponential family distribution of the form (3), with sufficient statistic $B(Z) = \frac{Z}{\sigma}$, and base measure $C(Z) = -\frac{Z^2}{2\sigma^2}$. Substituting these in (10), we get the distribution:

$$P(X;\theta) \propto \exp\left\{\sum_{r\in V}\frac{1}{\sigma_r}\theta_r X_r + \sum_{(r,t)\in E}\frac{1}{\sigma_r\sigma_t}\theta_{rt}\, X_r\, X_t - \sum_{r\in V}\frac{X_r^2}{2\sigma_r^2}\right\}, \tag{12}$$

which can be seen to be the multivariate Gaussian distribution. Note that the set of parameters $\{\theta_{rt}\}_{(r,t)\in E}$ entails a precision matrix that needs to be positive definite for a valid probability distribution.

2.3 Ising Models

The Ising model (Wainwright and Jordan, 2008) in turn can be derived from the construction in Theorem 2 with the Bernoulli distribution as the univariate exponential family distribution. The Bernoulli distribution is a member of the exponential family of the form (3), with sufficient statistic $B(X) = X$, and base measure $C(X) = 0$, and with variables taking values in the set $\mathcal{X} = \{0, 1\}$. Substituting these in (10), we get the distribution:

$$P(X;\theta) = \exp\left\{\sum_{(r,t)\in E}\theta_{rt}\, X_r\, X_t - A(\theta)\right\} \tag{13}$$

where we have ignored the singleton term, i.e. set $\theta_r = 0$ for simplicity. The form of the multinomial graphical model, an extension of the Ising model, can also be represented by (10) and has been previously studied in Jalali et al. (2011) and others. The Ising model imposes no constraint on its parameters, $\{\theta_{rt}\}$, for normalizability, since there are finitely many configurations of the binary random vector X.

2.4 Poisson Graphical Models

Poisson graphical models are an interesting instance with the Poisson distribution as the univariate exponential family distribution. The Poisson distribution is a member of the exponential family of the form (3), with sufficient statistic $B(X) = X$ and $C(X) = -\log(X!)$,

and with variables taking values in the set $\mathcal{X} = \{0, 1, 2, ...\}$. Substituting these in (10), we get the following Poisson graphical model distribution:

$$P(X;\theta) = \exp\left\{ \sum_{r\in V} \big(\theta_r X_r - \log(X_r!)\big) + \sum_{(r,t)\in E} \theta_{rt}\, X_r\, X_t - A(\theta) \right\}. \tag{14}$$

For this Poisson family, with some calculation, it can be seen that the normalizability condition, $A(\theta) < +\infty$, entails $\theta_{rt} \leq 0 \;\forall\; r, t$. In other words, the Poisson graphical model can only capture *negative* conditional relationships between variables.

2.5 Exponential Graphical Models

Another interesting instance uses the exponential distribution as the univariate exponential family distribution, with sufficient statistic $B(X) = -X$ and $C(X) = 0$, and with variables taking values in $\mathcal{X} = \{0\} \cup \mathbb{R}^+$. Such exponential distributions are typically used for data describing inter-arrival times between events, among other applications. Substituting these in (10), we get the following exponential graphical model distribution:

$$P(X;\theta) = \exp\left\{ -\sum_{r\in V} \theta_r X_r - \sum_{(r,t)\in E} \theta_{rt}\, X_r\, X_t - A(\theta) \right\}. \tag{15}$$

To ensure that the distribution is valid and normalizable, so that $A(\theta) < +\infty$, we then require that $\theta_r > 0, \theta_{rt} \geq 0 \;\forall\; r, t$. Because of the negative sufficient statistic, this implies that the exponential graphical model can only capture *negative* conditional relationships between variables.

3. Statistical Guarantees on Learning Graphical Model Structures

In this section, we study the problem of learning the graph structure of an underlying exponential family MRF, given i.i.d. samples. Specifically, we assume that we are given n samples of random vector $X^{1:n} := \{X^{(i)}\}_{i=1}^n$, from a pairwise exponential family MRF,

$$P(X;\theta^*) = \exp\left\{ \sum_{r\in V} \theta^*_r X_r + \sum_{(r,t)\in E^*} \theta^*_{rt}\, X_r\, X_t + \sum_r C(X_r) - A(\theta^*) \right\}. \tag{16}$$

The goal in graphical model structure recovery is to recover the edges E^* of the underlying graph $G = (V, E^*)$. Following Meinshausen and Bühlmann (2006); Ravikumar et al. (2010); Jalali et al. (2011), we will approach this problem via neighborhood estimation: where we estimate the neighborhood of each node individually, and then stitch these together to form the global graph estimate. Specifically, if we have an estimate $\widehat{N}(r)$ for the true neighborhood $N^*(r)$, then we can estimate the overall graph structure as

$$\widehat{E} = \cup_{r\in V} \cup_{t\in\widehat{N}(r)} \{(r,t)\}. \tag{17}$$

Remark. Note that the node-neighborhood estimates $\widehat{N}(r)$ might not be symmetric (i.e. there may be a pair $(r, s) \in V \times V$, with $r \in \widehat{N}(s)$, but $s \notin \widehat{N}(r)$). The graph-structure

estimate in (17) provides one way to reconcile these neighborhood estimates; see Meinshausen and Bühlmann (2006) for some other ways to do so (though as they note, these different estimates have asymptotically identical sparsistency guarantees: given exponential convergence in the probability of node-neighborhood recovery to one, the probability that the node-neighborhood estimates are symmetric, and hence that the different "reconciling" graph estimates would become identical, also converges to one.)

The problem of graph structure recovery can thus be reduced to the problem of recovering the neighborhoods of all the nodes in the graph. In order to estimate the neighborhood of any node in turn, we consider the sparsity constrained conditional MLE. Note that given the joint distribution in (16), the conditional distribution of X_r given the rest of the nodes is reduced to a GLM and given by

$$P(X_r|X_{V\setminus r}) = \exp\left\{ X_r\Big(\theta^*_r + \sum_{t\in N^*(r)} \theta^*_{rt} X_t\Big) + C(X_r) - D\Big(\theta^*_r + \sum_{t\in N^*(r)} \theta^*_{rt} X_t\Big)\right\}. \quad (18)$$

Let $\theta^*(r)$ be a set of parameters related to the node-conditional distribution of node X_r, i.e. $\theta^*(r) = (\theta^*_r, \theta^*_{\setminus r}) \in \mathbb{R}\times\mathbb{R}^{p-1}$ where $\theta^*_{\setminus r} = \{\theta^*_{rt}\}_{t\in V\setminus r}$ be a zero-padded vector, with entries θ^*_{rt} for $t \in N^*(r)$ and $\theta^*_{rt} = 0$, for $t \notin N^*(r)$. In order to infer the neighborhood structure for each node X_r, we solve the ℓ_1 regularized conditional log-likelihood loss:

$$\underset{\theta(r)\in\Omega}{\text{minimize}} \left\{ \ell\big(\theta(r); X^{1:n}\big) + \lambda_n \|\theta_{\setminus r}\|_1 \right\} \quad (19)$$

where Ω is the parameter space in $\mathbb{R}\times\mathbb{R}^{p-1}$, and $\ell\big(\theta(r); X^{1:n}\big)$ is the conditional log-likelihood of the distribution (18):

$$\begin{aligned}\ell\big(\theta(r); X^{1:n}\big) &:= -\frac{1}{n}\log\prod_{i=1}^{n} P\big(X_r^{(i)}|X^{(i)}_{V\setminus r}, \theta(r)\big) \\ &= \frac{1}{n}\sum_{i=1}^{n}\Big\{ -X_r^{(i)}\Big(\theta_r + \langle\theta_{\setminus r}, X^{(i)}_{V\setminus r}\rangle\Big) + D\Big(\theta_r + \langle\theta_{\setminus r}, X^{(i)}_{V\setminus r}\rangle\Big)\Big\}.\end{aligned}$$

Note that the parameter space Ω might be restricted, and strictly smaller than $\mathbb{R}\times\mathbb{R}^{p-1}$; for Poisson graphical models, $\theta_{rt} \le 0$ for all $r, t \in V$ for instance.

Given the solution $\widehat{\theta}(r)$ of the M-estimation problem above, we then estimate the node-neighborhood of r as $\widehat{N}(r) = \{t \in V\setminus r : \widehat{\theta}_{rt} \neq 0\}$. In what follows, when we focus on a fixed node $r \in V$, we will overload notation, and use $\theta \in \mathbb{R}\times\mathbb{R}^{p-1}$ as the parameters of the conditional distribution, suppressing dependence on the node r.

3.1 Conditions

A key quantity in the analysis is the Fisher information matrix, $Q^*_r = \nabla^2\ell\big(\theta^*; X^{1:n}\big)$, which is the Hessian of the node-conditional log-likelihood. In the following, we again will simply use Q^* instead of Q^*_r where the reference node r should be understood implicitly. We also use $S = \{(r,t) : t \in N^*(r)\}$ to denote the true neighborhood of node r, and S^c to denote its complement. We use Q^*_{SS} to denote the $d\times d$ sub-matrix of Q^* indexed by S where d is the number of neighborhoods of node r again suppressing dependence on r. Our first two conditions, mirroring those in Ravikumar et al. (2010), are as follows.

(C1) (Dependency condition) There exists a constant $\rho_{\min} > 0$ such that $\lambda_{\min}(Q^*_{SS}) \geq \rho_{\min}$ so that the sub-matrix of Fisher information matrix corresponding to true neighborhood has bounded eigenvalues. Moreover, there exists a constant $\rho_{\max} < \infty$ such that $\lambda_{\max}(\frac{1}{n}\sum_{i=1}^n [X^{(i)}_{V\setminus r}(X^{(i)}_{V\setminus r})^T]) \leq \rho_{\max}$.

These condition can be understood as ensuring that variables do not become overly dependent. We will also need an incoherence or irrepresentable condition on the Fisher information matrix as in Ravikumar et al. (2010).

(C2) (Incoherence condition) There exists a constant $\alpha > 0$, such that $\max_{t\in S^c} \|Q^*_{tS}(Q^*_{SS})^{-1}\|_1 \leq 1 - \alpha$.

This condition, standard in high-dimensional analyses, can be understood as ensuring that irrelevant variables do not exert an overly strong effect on the true neighboring variables.

A key technical facet of the linear, logistic, and multinomial models in Meinshausen and Bühlmann (2006); Ravikumar et al. (2010); Jalali et al. (2011), used heavily in their proofs, was that the random variables $\{X_r\}$ there were bounded with high probability. Unfortunately, in the general exponential family distribution in (18), we cannot assume this explicitly. Nonetheless, we show that we can analyze the corresponding regularized M-estimation problems under the following mild conditions on the log-partition functions of the joint and node-conditional distributions.

(C3) (Bounded Moments) For all $r \in V$, the first and second moments are bounded, so that

$$\mathbb{E}[X_r] \leq \kappa_m \quad \text{and} \quad \mathbb{E}[X_r^2] \leq \kappa_v,$$

for some constants κ_m, κ_v. Further, the log-partition function $A(\cdot)$ of the joint distribution (16) satisfies:

$$\max_{u:|u|\leq 1} \frac{\partial^2}{\partial \theta_r^2} A(\theta^* + ue_r) \leq \kappa_h,$$

for some constant κ_h, and where $e_r \in \mathbb{R}^{p^2}$ is an indicator vector that is equal to one at the index corresponding to θ_r, and zero everywhere else. Further, it holds that

$$\max_{\eta:|\eta|\leq 1} \frac{\partial^2}{\partial \eta^2} \bar{A}_r(\eta;\theta^*) \leq \kappa_h,$$

where $\bar{A}_r(\eta;\theta^*)$ is a slight variant of (11):

$$\bar{A}_r(\eta;\theta) := \log \int_X \exp\left\{\eta X_r^2 + \sum_{u\in V} \theta_u X_u + \sum_{(u,t)\in V^2} \theta_{ut}\, X_u\, X_t + \sum_{u\in V} C(X_u)\right\} dX \qquad (20)$$

for some scalar variable η.

(C4) For all $r \in V$, the log-partition function $D(\cdot)$ of the node-wise conditional distribution (18) satisfies: there exist functions $\kappa_1(n,p)$ and $\kappa_2(n,p)$ (that depend on the exponential family) such that, for all $\theta \in \Theta$ and $X \in \mathcal{X}$, $|D''(a)| \leq \kappa_1(n,p)$ where $a \in \left[b, b+4\kappa_2(n,p)\max\{\log n, \log p\}\right]$ for $b := \theta_r + \langle \theta_{\backslash r}, X_{V\backslash r}\rangle$. Additionally, $|D'''(b)| \leq \kappa_3(n,p)$ for all $\theta \in \Theta$ and $X \in \mathcal{X}$. Note that $\kappa_1(n,p)$,$\kappa_2(n,p)$ and $\kappa_3(n,p)$ are functions that might be dependent on n and p, which affect our main theorem below.

Conditions (C3) and (C4) are the key technical components enabling us to generalize the analyses in Meinshausen and Bühlmann (2006); Ravikumar et al. (2010); Jalali et al. (2011) to the general exponential family case. It is also important to note that almost all exponential family distributions including all our previous examples can satisfy (C4) with mild functions $\kappa_1(n,p)$, $\kappa_2(n,p)$ and $\kappa_3(n,p)$, as we will explicitly show later in this section. Comparing to the assumption in Yang et al. (2012) that requires $\|\theta^*\|_2 \leq 1$ for some exponential families, this will be much less restrictive condition on the minimum values of θ^* permitted to achieve variable selection consistency.

3.2 Statement of the Sparsistency Result

Armed with the conditions above, we can show that the random vector X following a exponential family MRF distribution in (16) is suitably well-behaved:

Proposition 3 *Suppose X is a random vector with the distribution specified in* (16)*. Then, for $\forall r \in V$,*

$$P\left(\frac{1}{n}\sum_{i=1}^{n}\left(X_r^{(i)}\right)^2 \geq \delta\right) \leq \exp\left(-c\, n\, \delta^2\right)$$

where $\delta \leq \min\{2\kappa_v/3, \kappa_h + \kappa_v\}$, and c is a positive constant.

We recall the notation that the superscript indicates the sample and the subscript indicates the node; so that $X^{(i)}$ is the i-th sample, while $X_s^{(i)}$ is the s-th variable/node of this random vector.

Proposition 4 *Suppose X is a random vector with the distribution specified in* (16)*. Then, for $\forall r \in V$,*

$$P\Big(|X_r| \geq \delta \log \eta\Big) \leq c\eta^{-\delta}$$

where δ is any positive real value, and c is a positive constant.

These propositions are key to the following sparsistency result for the general family of pairwise exponential family MRFs (16).

Theorem 5 *Consider a pairwise exponential family MRF distribution as specified in* (16)*, with true parameter θ^* and associated edge set E^* that satisfies Conditions (C1)-(C4). Suppose that $\min_{(s,t)\in E^*}|\theta^*_{rt}| \geq \frac{10}{\rho_{\min}}\sqrt{d}\lambda_n$, where d is the maximum neighborhood size. Suppose also that the regularization parameter is chosen such that $M_1\frac{(2-\alpha)}{\alpha}\sqrt{\kappa_1(n,p)}\sqrt{\frac{\log p}{n}} \leq \lambda_n \leq$*

$M_2\frac{(2-\alpha)}{\alpha}\kappa_1(n,p)\kappa_2(n,p)$ *for some constants* $M_1, M_2 > 0$. *Then, there exist positive constants* L, c_1, c_2 *and* c_3 *such that if* $n \geq Ld^2\kappa_1(n,p)(\kappa_3(n,p))^2 \log p(\max\{\log n, \log p\})^2$, *then with probability at least* $1 - c_1(\max\{n,p\})^{-2} - \exp(-c_2 n) - \exp(-c_3 n)$, *the following statements hold.*

(a) *(Unique Solution) For each node* $r \in V$, *the solution of the M-estimation problem in* (19) *is unique, and*

(b) *(Correct Neighborhood Recovery) The M-estimate also recovers the true neighborhood exactly, so that* $\widehat{N}(r) = N^*(r)$.

Note that if the neighborhood of each node is recovered with high probability, then by a simple union bound, the estimate in (17), $\widehat{E} = \cup_{r\in V} \cup_{t\in\widehat{N}(r)} \{(r,t)\}$ is equal to the true edge set E^* with high-probability.

In the following subsections, we investigate the consequences of Theorem 5 for the sparsistency of specific instances of our general exponential family MRFs.

3.3 Statistical Guarantees for Gaussian MRFs, Ising Models, Exponential Graphical Models

In order to apply Theorem 5 to a specific instance of our general exponential family MRFs, we need to specify the terms $\kappa_1(n,p)$, $\kappa_2(n,p)$ and $\kappa_3(n,p)$ defined in Condition (C4). It turns out that we can specify these terms for the Gaussian graphical models, Ising models and Exponential graphical model distributions, discussed in Section 2, in a similar manner, since the node-conditional log-partition function $D(\cdot)$ for all these distributions can be upper bounded by some constant independent of n and p. In particular, we can set $\kappa_1(n,p) := \kappa_1$, $\kappa_2(n,p) := \infty$ and $\kappa_1(n,p) := \kappa_3$ where κ_1 and κ_3 now become some constants depending on the distributions.

(Gaussian MRFs) Recall that the node-conditional distribution for Gaussian MRFs follow a univariate Gaussian distribution:

$$P(X_r|X_{V\setminus r}) \propto \exp\Big\{X_r\big(\theta_r + \sum_{t\in N(r)} \theta_{rt}X_t\big) - \frac{1}{2}X_r^2 - \frac{1}{2}\big(\theta_r + \sum_{t\in N(r)} \theta_{rt}X_t\big)^2\Big\}.$$

Note that following (Meinshausen and Bühlmann, 2006), we assume that $\sigma_r^2 = 1$ for all $r \in V$. The node-conditional log-partition function $D(\cdot)$ can thus be written as $D(\eta) := -\frac{1}{2}\eta^2$, so that $|D''(\eta)| = 1$ and $D'''(\eta) = 0$. We can thus set $\kappa_1 = 1$ and $\kappa_3 = 0$.

(Ising Models) For Ising models, node-conditional distribution follows a Bernoulli distribution:

$$P(X_r|X_{V\setminus r}) = \exp\Big\{X_r\Big(\sum_{t\in N(r)} \theta_{rt}X_t\Big) - \log\Big(1 + \exp\Big(\sum_{t\in N(r)} \theta_{rt}X_t\Big)\Big)\Big\}.$$

The node-conditional log-partition function $D(\cdot)$ can thus be written as $D(\eta) := \log\big(1 + \exp(\eta)\big)$, so that for any η, $|D''(\eta)| = \frac{\exp(\eta)}{(1+\exp(\eta))^2} \leq \frac{1}{4}$ and $|D'''(\eta)| = \big|\frac{\exp(\eta)(1-\exp(\eta))}{(1+\exp(\eta))^3}\big| < \frac{1}{4}$. Hence, we can set $\kappa_1 = \kappa_3 = 1/4$.

(Exponential Graphical Models) Lastly, for exponential graphical models, we have

$$P(X_r|X_{V\setminus r}) = \exp\Big\{ - X_r\Big(\theta_r + \sum_{t\in N(r)} \theta_{rt}X_t\Big) + \log\Big(\theta_r + \sum_{t\in N(r)} \theta_{rt}X_t\Big)\Big\}.$$

The node-conditional log-partition function $D(\cdot)$ can thus be written as $D(\eta) := -\log\eta$, with $\eta = \theta_r + \sum_{t\in N(r)} \theta_{rt}X_t$. Recall from Section 2.5 that the node parameters are strictly positive $\theta_r > 0$, and the edge-parameters are positive as well, $\theta_{rt} \geq 0$, as are the variables themselves $X_t \geq 0$. Thus, under the additional constraint that $\theta_r > a_0$ where a_0 is a constant smaller than θ_r^*, we have that $\eta := \theta_r + \sum_{t\in N(r)} \theta_{rt}X_t \geq a_0$. Consequently, $|D''(\eta)| = \frac{1}{\eta^2} \leq \frac{1}{a_0^2}$ and $|D'''(\eta)| = \big|\frac{2}{\eta^3}\big| \leq \frac{2}{a_0^3}$. We can thus set $\kappa_1 = \frac{1}{a_0^2}$ and $\kappa_3 = \frac{2}{a_0^3}$.

Armed with these derivations, we recover the following result on the sparsistency of Gaussian, Ising and Exponential graphical models, as a corollary of Theorem 5:

Corollary 6 *Consider a Gaussian MRF* (12) *or Ising model* (13) *or Exponential graphical model* (15) *distribution with true parameter θ^*, and associated edge set E^*, and which satisfies Conditions (C1)-(C3). Suppose that* $\min_{(s,t)\in E^*} |\theta^*_{rt}| \geq \frac{10}{\rho_{\min}}\sqrt{d}\lambda_n$. *Suppose also that the regularization parameter is set so that* $M\frac{(2-\alpha)}{\alpha}\sqrt{\kappa_1}\sqrt{\frac{\log p}{n}} \leq \lambda_n$ *for some constant $M > 0$. Then, there exist positive constants L, c_1, c_2 and c_3 such that if $n \geq L\kappa_1\kappa_3^2 d^2 \log p(\max\{\log n, \log p\})^2$, then with probability at least $1-c_1(\max\{n,p\})^{-2}-\exp(-c_2 n)$ $-\exp(-c_3 n)$, the statements on the uniqueness of the solution and correct neighborhood recovery, in Theorem 5 hold.*

Remarks. As noted, our models and theorems are quite general, extending well beyond the popular Ising and Gaussian graphical models. The graph structure recovery problem for Gaussian models was studied in Meinshausen and Bühlmann (2006) especially for the regime where the neighborhood sparsity index is *sublinear*, meaning that $d/p \to 0$. Besides the sublinear scaling regime, Corollary 6 can be adapted to entirely different types of scaling, such as the linear regime where $d/p \to \alpha$ for some $\alpha > 0$ (see Wainwright (2009) for details on adaptations to sublinear scaling regimes). Moreover, with κ_1 and κ_3 as defined above, Corollary 6 exactly recovers the result in Ravikumar et al. (2010) for the Ising models as a special case.

Also note that Corollary 6 provides tighter finite-sample bounds than the results of Yang et al. (2012). In particular, a sample size complexity necessary on λ_n to achieve sparsistent recovery here is $O(\sqrt{\frac{\log p}{n}})$, which is faster as compared to $O(\sqrt{\frac{\log p}{n^{1-\kappa}}})$ in Yang et al. (2012).

3.4 Statistical Guarantees for Poisson Graphical Models

We now consider the Poisson graphical model. Again, to derive the corresponding corollary of Theorem 5, we need to specify the terms $\kappa_1(n,p)$, $\kappa_2(n,p)$ and $\kappa_3(n,p)$ defined in Condition (C4). Recall that the node-conditional distribution of Poisson graphical models has the form:

$$P(X_r|X_{V\setminus r}) = \exp\Big\{X_r\Big(\theta_r + \sum_{t\in N(r)} \theta_{rt}X_t\Big) - \log(X_r!) - \exp\Big(\theta_r + \sum_{t\in N(r)} \theta_{rt}X_t\Big)\Big\}.$$

The node-conditional log-partition function $D(\cdot)$ can thus be written as $D(\eta) := \exp \eta$, with $\eta = \theta_r + \sum_{t \in N(r)} \theta_{rt} X_t$. Noting that the variables $\{X_t\}$ range over positive integers, and that feasible parameters θ_{rt} are negative, we obtain

$$\begin{aligned} D''(\eta) = D''\big(\theta_r + \langle \theta_{\backslash r},\, X_{V \backslash r} \rangle + 4\kappa_2(n,p) \log p'\big) &= \exp\big(\theta_r + \langle \theta_{\backslash r},\, X_{V \backslash r} \rangle + 4\kappa_2(n,p) \log p'\big) \\ &\leq \exp\big(\theta_r + 4\kappa_2(n,p) \log p'\big) \end{aligned}$$

where $p' = \max\{n, p\}$. Suppose that we restrict our attention on the subfamily where $\theta_r \leq a_0$ for some positive constant a_0. Then, if we choose $\kappa_2(n,p) := 1/(4 \log p')$, we then obtain $\theta_r + 4\kappa_2(n,p) \log p' \leq a_0 + 1$, so that setting $\kappa_1(n,p) := \exp(a_0 + 1)$ would satisfy Condition (C4). Similarly, we obtain $D'''\big(\theta_r + \langle \theta_{\backslash r},\, X_{V \backslash r} \rangle\big) = \exp\big(\theta_r + \langle \theta_{\backslash r},\, X_{V \backslash r} \rangle\big) \leq \exp(a_0 + 1)$, so that we can set $\kappa_3(n,p)$ to $\exp(a_0 + 1)$.

Armed with these settings, we recover the following corollary for Poisson graphical models:

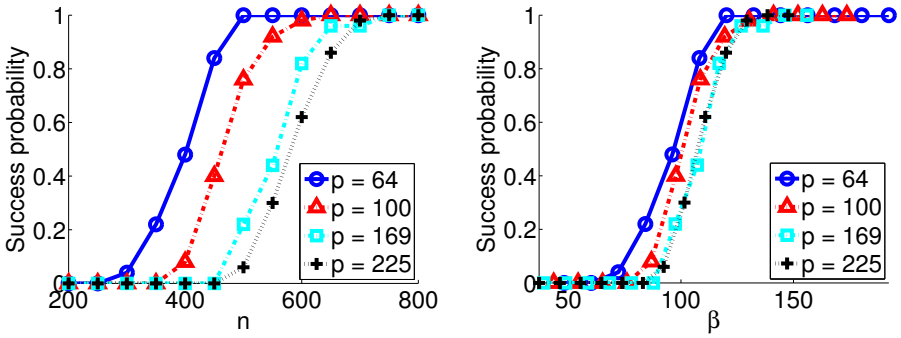
Corollary 7 *Consider a Poisson graphical model distribution as specified in* (14)*, with true parameters θ^*, and associated edge set E^*, that satisfies Conditions (C1)-(C3). Suppose that $\min_{(s,t) \in E^*} |\theta^*_{rt}| \geq \frac{10}{\rho_{\min}} \sqrt{d} \lambda_n$. Suppose also that the regularization parameter is chosen such that $M_1 \frac{(2-\alpha)}{\alpha} \sqrt{\kappa_1} \sqrt{\frac{\log p}{n}} \leq \lambda_n \leq M_2 \kappa_1 \frac{(2-\alpha)}{\alpha} \frac{1}{\max\{\log n, \log p\}}$ for some constants $M_1, M_2 > 0$. Then, there exist positive constants L, c_1, c_2 and c_3 such that if $n \geq L d^2 \kappa_1 \kappa_3^2 \log p (\max\{\log n, \log p\})^2$, then with probability at least $1 - c_1(\max\{n,p\})^{-2} - \exp(-c_2 n) - \exp(-c_3 n)$, the statements on the uniqueness of the solution and correct neighborhood recovery, in Theorem 5 hold.*

4. Experiments

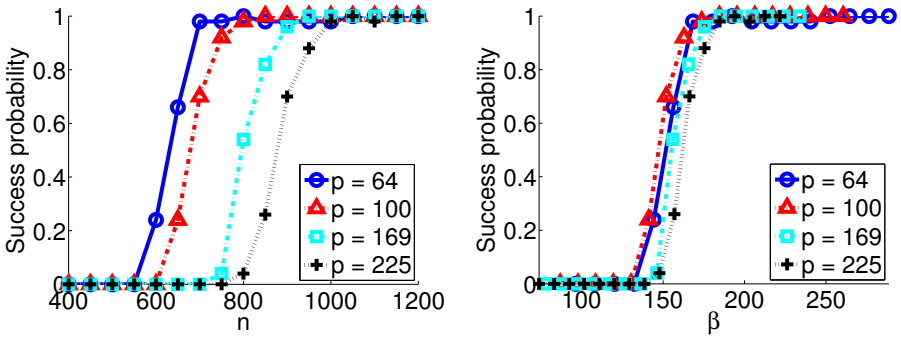
We evaluate our M-estimators for exponential family graphical models, specifically for the Poisson and exponential distributions, through simulations and real data examples. Neighborhood selection was performed for each M-estimator with an ℓ_1 penalty to induce sparsity and non-negativity or non-positivity constraints to enforce appropriate restrictions on the parameters. Optimization algorithms were implemented using projected gradient descent (Daubechies et al., 2008; Beck and Teboulle, 2010), which since the objectives are convex, is guaranteed to converge to the global optimum. Further details on the optimization problems used for our M-estimators are given in the Appendix E.

4.1 Simulation Studies

We provide a small simulation study that corroborates our sparsistency results; specifically Corollary 6 for the exponential graphical model, where node-conditional distributions follow an exponential distribution, and Corollary 7 for the Poisson graphical model, where node-conditional distributions follow a Poisson distribution. We instantiated the corresponding exponential and Poisson graphical model distributions in (15) and (14) for 4 nearest neighbor lattice graphs ($d = 4$), with varying number of nodes, $p \in \{64, 100, 169, 225\}$, and with identical edge weights for all edges: for exponential MRF, $\theta^*_r = 0.1$ and $\theta^*_{rt} = 1$, and, for Poisson MRF, $\theta^*_r = 2$ and $\theta^*_{rt} = -0.1$. We generated i.i.d. samples from these distributions using Gibbs sampling, and solved our sparsity-constrained M-estimation problem by setting



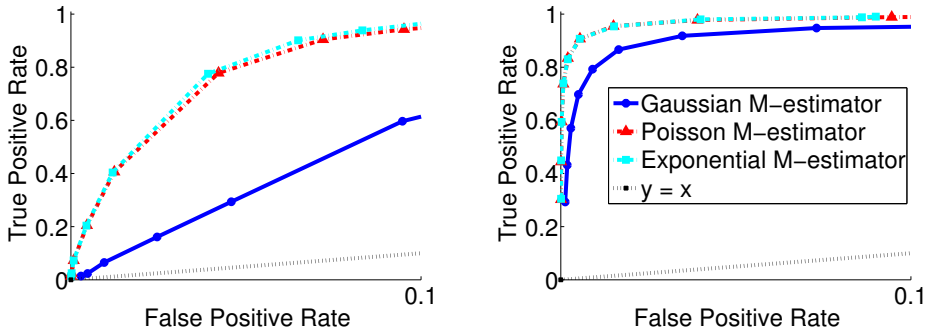
(a) Exponential Grid Structure



(b) Poisson Grid Structure

Figure 1: Probabilities of successful support recovery for the (a) exponential MRF, grid structure with parameters $\theta^*_r = 0.1$ and $\theta^*_{rt} = 1$, and the (b) Poisson MRF, grid structure with parameters $\theta^*_r = 2$ and $\theta^*_{rt} = -0.1$. The empirical probability of successful edge recovery over 50 replicates is shown versus the sample size n (left), and verses the re-scaled sample size $\beta = n/(\log p)$ (right). The empirical curves align for the latter, thus verifying the logarithmic dependence of n on p as obtained in our sparsistency analysis.

$\lambda_n = c\sqrt{\frac{\log p}{n}}$, following our corollaries; $c = 3$ for exponential MRF, and 15 for Poisson MRF. We repeated each simulation 50 times and measured the empirical probability over the 50 trials that our penalized graph estimate in (17) successfully recovered all edges, that is, $P(\hat{E} = E^*)$. The left panels of Figure 1(a) and Figure 1(b) show the empirical probability of successful edge recovery. In the right panel, we plot the empirical probability against a re-scaled sample size $\beta = n/(\log p)$. According to our corollaries, the sample size n required for successful graph structure recovery scales logarithmically with the number of nodes p. Thus, we would expect the empirical curves for different problem sizes to more closely align with this re-scaled sample size on the horizontal axis, a result clearly seen in the right panels of Figure 1. This small numerical study thus corroborates our theoretical sparsistency results.



(a) Exponential Grid Structure (b) Poisson Grid Structure

Figure 2: Receiver-operator curves (ROC) computed by varying the regularization parameter, λ_n. High-dimensional data is generated according to (a) the Exponential MRF with $(n, p) = (150, 225)$ and to (b) the Poisson MRF with $(n, p) = (100, 225)$. Results are compared for three M-estimators: that of the Poisson, exponential, and Gaussian distributions.

We also evaluate the comparative performance of our M-estimators for recovering the true edge structure from the different types of data. Specifically, we consider the three typical examples in our unified neighborhood selection approach: the Poisson M-estimator, the Exponential M-estimator, and the well-known Gaussian M-estimator by (Meinshausen and Bühlmann, 2006). In order to extensively compare their performances, we compute the receiver-operator-curves for the overall graph recovery by varying the regularization parameter, λ_n. In Figure 2, the same graph structures for the exponential MRF ($\theta^*_r = 0.1$ and $\theta^*_{rt} = 1$) and the Poisson MRF ($\theta^*_r = 2$ and $\theta^*_{rt} = -0.1$) with 4 nearest neighbors, are used as in the previous simulation. Moreover, we focus on the high-dimensional regime where $n < p$. As shown in the figure, exponential and Poisson M-estimators outperform and have significant advantage over Gaussian neighborhood selection approach if the data is generated according to exponential or Poisson MRFs. One interesting phenomenon we observe is that exponential and Poisson M-estimators perform similarly regardless of the underlying graphical model distribution. This likely occurs as our estimator maximizes the conditional likelihoods by fitting penalized GLMs. Note that GLMs assume that the conditional mean of the regression model follows an exponential family distribution. As both the Poisson distribution and the exponential distribution have the same mean, the rate parameter, λ, we would expect GLM-based methods that fit conditional means to perform similarly.

As discussed at end of Section 2, the exponential and Poisson graphical models are able to capture only negative conditional dependencies between random variables, and our corresponding M-estimators are computed under this constraint. In our last simulation, we evaluate the impact of this restriction when the true graph contains both positive and negative edge weights. As there does not exist a proper MRF related to the Poisson and exponential distributions with both positive and negative dependencies, we resort to gen-

erating data from via a copula transform. In particular, we first generate multivariate Gaussian samples from $N(0, \Sigma)$ where $\Theta = \Sigma^{-1}$ is the precision matrix corresponding to the 4 nearest neighbor grid structure previously considered. Specifically, Θ has all ones on the diagonal and $\theta^*_{rt} = \pm 0.2$ with equal probabilities. We then use a standard copula transform to make the marginals of the generated data approximately Poisson. Figure 3 again present receiver operator curves (ROC) for the three different classes of M-estimators on the copula transformed data, transformed to the Poisson distribution. In the left of Figure 3, we consider *signed* support recovery where we define the true positive rate as $\frac{\#\text{ of edges s.t. } \text{sign}(\theta^*_{rt}) = \text{sign}(\widehat{\theta}_{rt})}{\#\text{ of edges}}$. In the right, on the other hand, we *ignore* the positive edges so that true positive rate is now $\frac{\#\text{ of edges s.t. } \text{sign}(\theta^*_{rt}) = \text{sign}(\widehat{\theta}_{rt}) = -1}{\#\text{ of negative edges}}$. Note that the false positive rate is also defined similarly. As expected, the results indicate that our Poisson and exponential M-estimators fail to recover the edges with positive conditional dependencies recovered by the Gaussian M-estimator. However, when attention is restricted to negative conditional dependencies, our method outperforms the Gaussian M-estimator. Notice also that for the exponential and Poisson M-estimators, the highest false positive rate achieved is around 0.15. This likely occurs due to the constraints enforced by our M-estimators that force the weights of potential positive conditional dependent edges to be zero. Thus, while the restrictions on the edge weights may be severe, for the purpose of estimating negative conditional dependencies with limited false positives, the Poisson and exponential M-estimators have an advantage.

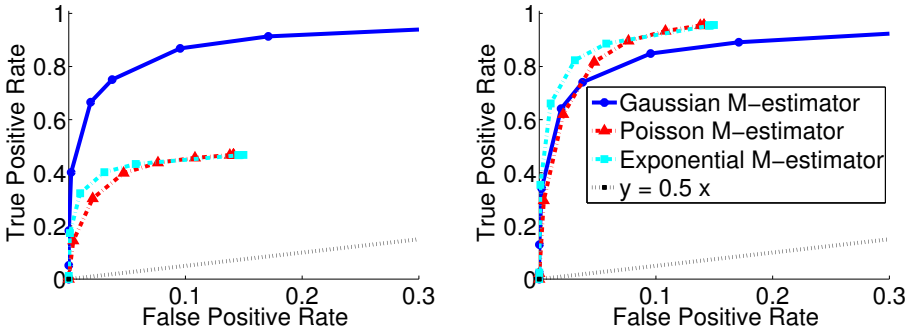


Figure 3: Receiver-operator curves (ROC) computed by varying the regularization parameter, λ_n, for data, $(n, p) = (200, 225)$, generated via Poisson copula transform according to a network with both positive and negative conditional dependencies. Left plot denotes results on overall edge recovery, while right plot denotes recovery of the edges with negative weights corresponding to negative conditional dependencies.

4.2 Real Data Examples

To demonstrate the versatility of our family of graphical models, we also provide two real data examples: a meta-miRNA inhibitory network estimated by the Poisson graphical model, Figure 4, and a cell signaling network estimated by the exponential graphical model, Figure 5.

When applying our family of graphical models, there is always a question of whether our model is an appropriate fit for the observed data. Typically, one can assess model fit using goodness-of-fit tests. For the Gaussian graphical model, this reduces to testing whether the data follows a multivariate Gaussian distribution. For general exponential family graphical models, testing for goodness-of-fit is more challenging. Some have proposed likelihood ratio tests specifically for lattice systems with a fixed and known dependence structure (Besag, 1974). When the network structure is unknown, however, there are no such existing tests. While we leave the development of an exact test to future work, we provide a heuristic that can help us understand whether our model is appropriate for a given dataset.

Recall that our model assumes that conditional on its node-neighbors, each variable is distributed according to an exponential family. Thus, if the neighborhood is known, our conditional models are simply GLMs, for which the goodness-of-fit can be assessed compared to a null model by a likelihood ratio test (McCullagh and Nelder, 1989). When neighborhoods must be estimated, and specifically when estimated via an ℓ_1-norm penalty, the resulting ratio of likelihoods no longer follow a chi-squared distribution (Bühlmann, 2011). Recently, for the ℓ_1 linear regression case, Lockhart et al. (2014) have shown that the difference in the residual sums of squares follows an exponential distribution. Similar results have not yet been extended to the penalized GLM case. In the absence of such tests, we propose a simple heuristic: for each node, first estimate the node-neighborhood via our proposed M-estimator. Next, assuming the neighborhood is fixed, fit a GLM and compare the fit of this model to that of a null model (only an intercept term) via the likelihood ratio test. One can then heuristically assess the overall goodness-of-fit by examining the fit of a GLM to all the nodes. This procedure is clearly not an exact test, and following from Lockhart et al. (2014), it is likely conservative. In the absence of an exact test, which we leave for future work, this heuristic provides some assurances about the appropriateness of our model for real data.

4.2.1 Poisson Graphical Model: Meta-miRNA Inhibitory Network

Gaussian graphical models have often been used to study high-throughput genomic networks estimated from microarray data (Pe'er et al., 2001; Friedman, 2004; Wei and Li, 2007). Many high-throughput technologies, however, do not produce even approximately Gaussian data, so that our class of graphical models could be particularly important for estimating genomic networks from such data. We demonstrate the applicability of our class of models by estimating a meta-miRNA inhibitory network for breast cancer estimated by a Poisson graphical model. Level III breast cancer miRNA expression (Cancer Genome Atlas Research Network, 2012) as measured by next generation sequencing was downloaded from the TCGA portal (http://tcga-data.nci.nih.gov/tcga/). MicroRNAs (miRNA) are short RNA fragments that are thought to be post-transcriptional regulators, predominantly inhibiting translation. Measuring miRNA expression by high-throughput sequencing results

in *count data* that is zero-inflated, highly skewed, and whose total count volume depends on experimental conditions (Li et al., 2011). Data was processed to be approximately Poisson by following the steps described in (Allen and Liu, 2013). In brief, the data was quantile corrected to adjust for sequencing depth (Bullard et al., 2010); the miRNAs with little variation across the samples, the bottom 50%, were filtered out; and the data was adjusted for possible over-dispersion using a power transform and a goodness of fit test (Li et al., 2011). We also tested for batch effects in the resulting data matrix consisting of 544 subjects and 262 miRNAs: we fit a Poisson ANOVA model (Leek et al., 2010), and only found 4% of miRNAs to be associated with batch labels; and thus no significant batch association was detected. As several miRNAs likely target the same gene or genes in the same pathway, we expect there to be strong positive dependencies among variables that cannot be captured directly by our Poisson graphical model which only permits negative conditional relationships. Thus, we will use our model to study inhibitory relationships between what we term meta-miRNAs, or groups of miRNAs that are tightly positively correlated. To accomplish this, we further processed our data to form clusters of positively correlated miRNAs using hierarchical clustering with average linkage and one minus the correlation as the distance

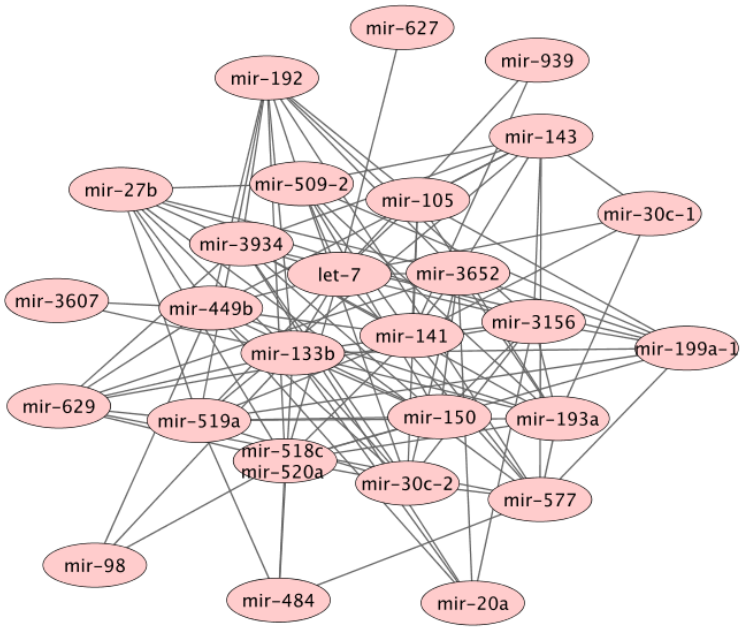


Figure 4: Meta-miRNA inhibitory network for breast cancer estimated via Poisson graphical models from miRNA-sequencing data. Level III data from TCGA was processed into tightly correlated clusters, meta-miRNAs, with the driver miRNAs identified for each cluster taken as the set of nodes for our network. The Poisson network reveals major inhibitory relationships between three hub miRNAs, two of which have been previously identified as tumor suppressors in breast cancer.

metric. This resulted in 40 clusters of tightly positively correlated miRNAs. The nodes of our meta-miRNA network were then taken as a the medoid, or median centroid defined as the miRNA closest in Euclidean distance to the cluster centroid, in each group.

A Poisson graphical model was fit to the meta-miRNA data by performing neighborhood selection with the sparsity of the graph determined by stability selection (Liu et al., 2010). The heuristic previously discussed was used to assess goodness-of-fit for our model. Out of the 40 node-neighborhoods tested via a likelihood ratio test, 36 exhibited p-values less than 0.05, and 34 were less than 0.05/40, the Bonferroni-adjusted significance level. These results show that the Poisson GLM is a significantly better fit for the majority of node-neighborhoods than the null model, indicating that our Poisson graphical model is appropriate for this data. The results of our estimated Poisson graphical model, Figure 4 (left), are consistent with the cancer genomics literature. First, the meta-miRNA inhibitory network has three major hubs. Two of these, miR-519 and miR-520, are known to be breast cancer tumor suppressors, suppressing growth by reducing HuR levels (Abdelmohsen et al., 2010) and by targeting NF-KB and TGF-beta pathways (Keklikoglou et al., 2012) respectively. The third major hub, miR-3156, is a miRNA of unknown function; from its major role in our network, we hypothesize that miR-3156 is also associated with tumor suppression. Also interestingly, let-7, a well-known miRNA involved in tumor metastasis (Yu et al., 2007), plays a central role in our network, sharing edges with the five largest hubs. This suggests that our Poisson graphical model has recovered relevant negative relationships between miRNAs with the five major hubs acting as suppressors, and the central let-7 miRNA and those connected to each of the major hubs acting as enhancers of tumor progression in breast cancer.

4.2.2 Exponential Graphical Model: Inhibitory Cell-Signaling Network

We demonstrate our exponential graphical model, derived from the univariate exponential distribution, using a protein signaling example (Sachs et al., 2005). Multi-florescent flow cytometry was used to measure the presence of eleven proteins ($p = 11$) in $n = 7462$ cells. This data set was first analyzed using Bayesian Networks in Sachs et al. (2005) and then using the graphical lasso algorithm in Friedman et al. (2007). Measurements from flow-cytometry data typically follow a left skewed distribution. Thus to model such data, these measurements are typically normalized to be approximately Gaussian using a log transform after shifting the data to be non-negative (Herzenberg et al., 2006). Here, we demonstrate the applicability of our exponential graphical models to recover networks directly from continuous skewed data, so that we learn the network directly from the flow-cytometry data without any log or such transforms. Our pre-processing is limited to shifting the data for each protein so that it consists of non-negative values. For comparison purposes, we also fit a Gaussian graphical model to the log-transformed data.

We then learned an exponential and Gaussian graphical model from this flow cytometry data using stability selection (Liu et al., 2010) to select the sparsity of the graphs. The goodness-of-fit heuristic previously described was used to assess the appropriateness of our model. Out of the eight connected node-neighborhoods, the likelihood ratio test was statistically significant for seven neighborhoods, indicating that our exponential GLM is a better fit than the null model. The estimated protein-signaling network is shown on the

right in Figure 5 with that of the Gaussian graphical model fit to the log-transformed data on the left. Estimated negative conditional dependencies are shown in red. Recall that the exponential graphical model restricts the edge weights to be non-negative; because of the negative inverse link, this implies that only negative conditional associations can be estimated. Notice that our exponential graphical model finds that PKA, protein kinase A, is a major protein inhibitor in cell signaling networks. This is consistent with the inhibitory relationship of PKA as estimated by the Gaussian graphical model, right Figure 5, as well as its hub status in the Bayesian network of (Sachs et al., 2005). Interestingly, our exponential graphical model also finds a clique between PIP2, Mek, and P38, which was not found by Gaussian graphical models.

5. Discussion

We study what we call the class of exponential family graphical models that arise when we assume that node-wise conditional distributions follow exponential family distributions. Our work broadens the class of off-the-shelf graphical models from classical instances such as Ising and Gaussian graphical models. In particular, our class of graphical models provide

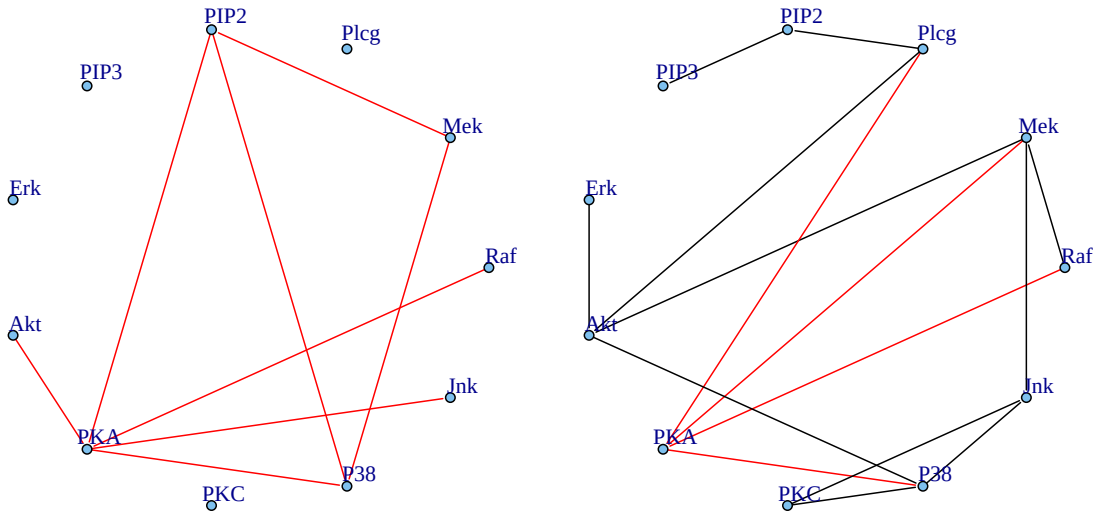


Figure 5: Cell signaling network estimated from flow cytometry data via exponential graphical models (left) and Gaussian graphical models (right). The exponential graphical model was fit to un-transformed flow cytometry data measuring 11 proteins, and the Gaussian graphical model to log-transformed data. Estimated negative conditional dependencies are given in red. Both networks identify PKA (protein kinase A) as a major inhibitor, consistent with previous results.

closed form multivariate densities as extensions of several univariate exponential family distributions (e.g. Poisson, exponential, negative binomial) where few currently exist; and thus may be of further interest to the statistical community. Further, we provide simple M-estimators for estimating any of these graphical models from data, by fitting node-wise penalized conditional exponential family distributions, and show that these estimators enjoy strong statistical guarantees. The statistical analyses of our M-estimators required subtle techniques that may be of general interest in the analysis of sparse M-estimation.

There are many avenues of future work related to our proposed models. We assume that all conditional distributions are members of an exponential family. To determine whether this assumption is appropriate in practice for real data, a goodness-of-fit procedure is needed. While we have proposed a heuristic to this effect, more work is needed to determine a rigorous likelihood ratio test for testing model fit. For several instances of our proposed class of models, specifically those with variables with infinite domains, severe restrictions on the parameter space are sometimes needed. For instance, the Poisson and exponential graphical models studied in Section 4, could only model negative conditional dependencies, which may not always be desirable in practice. A key question for future work is whether these restrictions can be relaxed for particular exponential family distributions. Finally, while we have focused on single parameter exponential families, it would be interesting to investigate the consequences of using multi-parameter exponential family distributions. Overall, our work has opened avenues for learning Markov networks from a broad class of univariate distributions, the properties and applications of which leave much room for future research.

Acknowledgments

We would like to acknowledge support for this project from ARO W911NF-12-1-0390, NSF IIS-1149803, IIS-1320894, IIS-1447574, and DMS-1264033 (PR and EY); NSF DMS-1264058 and DMS-1209017 (GA); and the Houston Endowment and NSF DMS-1263932 (ZL).

Appendix A. Proof of Theorem 2

The proof follows the development in Besag (1974), where they consider the case with $k = 2$. We define $Q(X)$ as $Q(X) := \log(P(X)/P(\mathbf{0}))$, for any $X = (X_1, \ldots, X_p) \in \mathcal{X}^p$ where $P(\mathbf{0})$) denotes the probability that all random variables take 0. Given any X, also denote $\overline{X}_{r:0} := (X_1, \ldots, X_{r-1}, 0, X_{r+1}, \ldots, X_p)$. Now, consider the following the most general form for $Q(X)$:

$$Q(X) = \sum_{1 \le r \le p} X_r G_r(X_r) + \ldots + \sum_{1 \le r_1 < r_2 < \ldots < r_k \le p} X_{r_1} \ldots X_{r_k} G_{r_1 \ldots r_k}(X_{r_1}, \ldots, X_{r_k}), \quad (21)$$

since the joint distribution has factors of at most size k. By the definition of Q and some algebra (See Section 2 of Besag (1974) for details), it can then be seen that

$$\begin{aligned} &\exp(Q(X) - Q(\overline{X}_{r:0})) \\ &= P(X_r | X_1, \ldots, X_{r-1}, X_{r+1}, \ldots, X_p) / P(0 | X_1, \ldots, X_{r-1}, X_{r+1}, \ldots, X_p). \end{aligned} \quad (22)$$

Now, consider the simplifications of both sides of (22). For notational simplicity, we fix $r = 1$ for a while. Given the form of $Q(X)$ in (21), we have

$$Q(X) - Q(\overline{X}_{1:0}) = X_1\Bigg(G_1(X_1) + \sum_{2\le t\le p} X_t G_{1t}(X_1, X_t) + \ldots + \sum_{2\le t_2<t_3<\ldots<t_k\le p} X_{t_2}\ldots X_{t_k} G_{1t_2\ldots t_k}(X_1, X_{t_2}\ldots, X_{t_k})\Bigg). \tag{23}$$

By given the exponential family form of the node-conditional distribution specified in the statement, right-hand side of (22) become

$$\log\frac{P(X_1|X_2,\ldots,X_p)}{P(0|X_2,\ldots,X_p)} = E(X_{V\setminus 1})(B(X_1) - B(0)) + (C(X_1) - C(0)). \tag{24}$$

Setting $X_t = 0$ for all $t \neq 1$ in (23) and (24), we obtain

$$X_1 G_1(X_1) = E(\mathbf{0})(B(X_1) - B(0)) + (C(X_1) - C(0)).$$

Setting $X_{t_2} = 0$ for all $t_2 \notin \{1, t\}$,

$$X_1 G_1(X_1) + X_1 X_t G_{1t}(X_1, X_t) = E(0,\ldots,X_t,\ldots,0)(B(X_1) - B(0)) + (C(X_1) - C(0)).$$

Recovering the index 1 back to r yields

$$X_r G_r(X_r) = E(\mathbf{0})(B(X_r) - B(0)) + (C(X_r) - C(0)),$$
$$X_r G_r(X_r) + X_r X_t G_{rt}(X_r, X_t) = E(0,\ldots,X_t,\ldots,0)(B(X_r) - B(0)) + (C(X_r) - C(0)).$$

Similarly,

$$X_t G_t(X_t) + X_r X_t G_{rt}(X_r, X_t) = E(0,\ldots,X_r,\ldots,0)(B(X_t) - B(0)) + (C(X_t) - C(0)). \tag{25}$$

From the above three equations, we obtain

$$X_r X_t G_{rt}(X_r, X_t) = \theta_{rt}(B(X_r) - B(0))(B(X_t) - B(0)).$$

More generally, by considering non-zero triplets, and setting $X_v = 0$ for all $v \notin \{r, t, u\}$, we obtain

$$X_r G_r(X_r) + X_r X_t G_{rt}(X_r, X_t) + X_r X_u G_{ru}(X_r, X_u) + X_r X_t X_u G_{rtu}(X_r, X_t, X_u) = E(0,\ldots,X_t,\ldots,X_u,\ldots,0)(B(X_r) - B(0)) + (C(X_r) - C(0)), \tag{26}$$

so that by a similar reasoning we can obtain

$$X_r X_t X_u G_{rtu}(X_r, X_t, X_u) = \theta_{rtu}(B(X_r) - B(0))(B(X_t) - B(0))(B(X_u) - B(0)).$$

More generally, we can show that

$$X_{t_1}\ldots X_{t_k} G_{t_1,\ldots,t_k}(X_{t_1},\ldots,X_{t_k}) = \theta_{t_1,\ldots,t_k}(B(X_{t_1}) - B(0))\ldots(B(X_{t_k}) - B(0)).$$

Thus, the k-th order factors in the joint distribution as specified in (21) are tensor products of $(B(X_r) - B(0))$, thus proving the statement of the theorem.

Appendix B. Proof of Proposition 3

By the definition of (20) with the following simple calculation, the moment generating function of X_r^2 becomes

$$\begin{aligned}\log \mathbb{E}[\exp(aX_r^2)] &= \log \int_X \exp\Big\{aX_r^2 + \sum_{t\in V}\theta_t^* X_t + \sum_{(t,u)\in E}\theta_{tu}^* X_t X_u + \sum_{t\in V} C(X_t) - A(\theta^*)\Big\} \\ &= \bar{A}_r(a;\theta^*) - \bar{A}_r(0;\theta^*).\end{aligned}$$

Suppose that $a \le 1$. Then, by a Taylor Series expansion, we have for some $\nu \in [0,1]$

$$\bar{A}_r(a;\theta^*) - \bar{A}_r(0;\theta^*) = a\frac{\partial}{\partial\eta}\bar{A}_r(0;\theta^*) + \frac{1}{2}a^2\frac{\partial^2}{\partial\eta^2}\bar{A}_r(\nu a;\theta^*) \le \kappa_v a + \frac{1}{2}\kappa_h a^2,$$

where the inequality uses Condition (C3). Note that since the derivative of log-partition function is the mean of the corresponding sufficient statistics and $\bar{A}_r(0;\theta) = A(\theta)$, $\frac{\partial}{\partial\eta}\bar{A}_r(0;\theta^*) = \mathbb{E}[X_r^2] \le \kappa_v$ by assumption. Thus, by the standard Chernoff bounding technique, for all positive $a \le 1$,

$$P\bigg(\frac{1}{n}\sum_{i=1}^n \big(X_r^{(i)}\big)^2 \ge \delta\bigg) \le \exp(-n\delta a + n\kappa_v a + \frac{n}{2}\kappa_h a^2).$$

With the choice of $a = \frac{\delta-\kappa_v}{\kappa_h} \le 1$, we obtain

$$P\bigg(\frac{1}{n}\sum_{i=1}^n \big(X_r^{(i)}\big)^2 \ge \delta\bigg) \le \exp\bigg(-n\frac{(\delta-\kappa_v)^2}{2\kappa_h}\bigg) \le \exp\Big(-n\frac{\delta^2}{8\kappa_h}\Big),$$

provided that $\delta \le 2\kappa_v/3$, as in the statement.

Appendix C. Proof of Proposition 4

Let $\bar{v} \in \mathbb{R}^{p+\binom{p}{2}}$ be the zero-padded parameter with only one non-zero coordinate, which is 1, for the sufficient statistics X_r so that $\|\bar{v}\|_2 = 1$. A simple calculation shows that

$$\log \mathbb{E}[\exp(X_r)] = A(\theta^* + \bar{v}) - A(\theta^*).$$

By a Taylor Series expansion and Condition (C3), we have for some $\nu \in [0,1]$

$$\begin{aligned}A(\theta^* + \bar{v}) - A(\theta^*) &= \nabla A(\theta^*)\cdot\bar{v} + \frac{1}{2}\bar{v}^T\nabla^2 A(\theta^* + \nu\bar{v})\bar{v} \\ &\overset{(i)}{\le} \mathbb{E}[X_r]\|\bar{v}\|_2 + \frac{1}{2}\frac{\partial^2}{\partial\theta_r^2}A(\theta^* + \nu\bar{v})\|\bar{v}\|_2^2 \le \kappa_m + \frac{1}{2}\kappa_h\end{aligned}$$

where the inequality (i) uses the fact that $\bar{v}$ has only nonzero element for the sufficient statistics X_r. Thus, again by the standard Chernoff bounding technique, for any positive constant a, $P(X_r \ge a) \le \exp(-a + \kappa_m + \frac{1}{2}\kappa_h)$, and by setting $a = \delta\log\eta$ we have

$$P(X_r \ge \delta\log\eta) \le \exp(-\delta\log\eta + \kappa_m + \frac{1}{2}\kappa_h) \le c\eta^{-\delta}$$

where $c = \exp(\kappa_m + \frac{1}{2}\kappa_h)$, as claimed.

Appendix D. Proof of Theorem 5

In this section, we sketch the proof of Theorem 5 following the *primal-dual witness* proof technique in Wainwright (2009); Ravikumar et al. (2010). We first note that the optimality condition of the convex program (19) can be written as

$$\nabla \ell(\widehat{\theta}; X^{1:n}) + \lambda_n \widehat{Z} = 0 \tag{27}$$

where $\widehat{Z}$ is a length p vector: $\widehat{Z}_{\backslash r} \in \partial \|\widehat{\theta}_{\backslash r}\|_1$ is a length $(p-1)$ subgradient vector where $\widehat{Z}_{rt} = \text{sign}(\widehat{\theta}_{rt})$ if $\widehat{\theta}_{rt} \neq 0$, and $|\widehat{Z}_{rt}| \leq 1$ otherwise; while $\widehat{Z}_r$, corresponding to θ_r, is set to 0 since the nodewise term θ_r is not penalized in the M-estimation problem (19).

Note that in a high-dimensional regime with $p \gg n$, the convex program (19) is not necessarily *strictly* convex, so that it might have multiple optimal solutions. However, the following lemma, adapted from Ravikumar et al. (2010), shows that nonetheless the solutions share their support set under certain conditions. We first recall the notation $S = \{(r,t) : t \in N^*(r)\}$ to denote the true neighborhood of node r, and S^c to denote its complement.

Lemma 8 *Suppose that there exists a primal optimal solution $\widehat{\theta}$ with associated subgradient $\widehat{Z}$ such that $\|\widehat{Z}_{S^c}\|_\infty < 1$. Then, any optimal solution $\tilde{\theta}$ will satisfy $\tilde{\theta}_{S^c} = 0$. Moreover, under the condition of $\|\widehat{Z}_{S^c}\|_\infty < 1$, if Q^*_{SS} is invertible, then $\widehat{\theta}$ is the unique optimal solution of* (19).

Proof This lemma can be proved by the same reasoning developed for the special cases Wainwright (2009); Ravikumar et al. (2010) in our framework; As in the previous works, for any node-conditional distribution in the form of exponential family, we are solving the convex objective with ℓ_1 regularizer (19). Therefore, the problem can be written as an equivalent constrained optimization problem, and by the complementary slackness, for any optimal solution $\tilde{\theta}$, we have $\langle \widehat{Z}, \tilde{\theta} \rangle = \|\tilde{\theta}\|_1$. This simply implies that for all index j for which $|\widehat{Z}_j| < 1$, $\tilde{\theta}_j = 0$ (See Ravikumar et al. (2010) for details). Therefore, if there exists a primal optimal solution $\widehat{\theta}$ with associated subgradient $\widehat{Z}$ such that $\|\widehat{Z}_{S^c}\|_\infty < 1$, then, any optimal solution $\tilde{\theta}$ will satisfy $\tilde{\theta}_{S^c} = 0$ as claimed.

Finally, we consider the restricted optimization problem subject to the constraint $\theta_{S^C} = 0$. For this restricted optimization problem, if the Hessian, Q^*_{SS}, is positive definite as assumed in the lemma, then, this restricted problem is strictly convex, and its solution is unique. Moreover, since all primal optimal solutions of (19), $\tilde{\theta}$, satisfy $\tilde{\theta}_{S^c} = 0$ as discussed, the solution of the restricted problem is the unique solution of (19). ■

We use this lemma to prove the theorem following the *primal-dual witness* proof technique in Wainwright (2009); Ravikumar et al. (2010). Specifically, we *explicitly construct* a pair $(\widehat{\theta}, \widehat{Z})$ as follows (denoting the true support set of the edge parameters by S):

(a) Recall that $\theta(r) = (\theta_r, \theta_{\backslash r}) \in \mathbb{R} \times \mathbb{R}^{p-1}$. We first fix $\theta_{S^c} = 0$ and solve the restricted optimization problem: $(\widehat{\theta}_r, \widehat{\theta}_S, 0) = \arg\min_{\theta_r \in \mathbb{R}, (\theta_S, 0) \in \mathbb{R}^{p-1}} \{\ell(\theta; X^{1:n}) + \lambda_n \|\theta_S\|_1\}$, and $\widehat{Z}_S = \text{sign}(\widehat{\theta}_S)$.

(b) We set $\widehat{\theta}_{S^c} = 0$.

(c) We set $\widehat{Z}_{S^c}$ to satisfy the condition (27) with $\widehat{\theta}$ and $\widehat{Z}_S$.

By construction, the support of $\widehat{\theta}$ is included in the true support S of θ^*, so that we would finish the proof of the theorem provided (a) $\widehat{\theta}$ satisfies the stationary condition of (19), as well as the condition $\|\widehat{Z}_{S^c}\|_\infty < 1$ in Lemma 8 with high probability, so that by Lemma 8, the primal solution $\widehat{\theta}$ is guaranteed to be unique; and (b) the support of $\widehat{\theta}$ is not strictly within the true support S. We term these conditions *strict dual feasibility*, and *sign consistency* respectively.

We will now rewrite the subgradient optimality condition (27) as

$$\nabla^2 \ell(\theta^*; X^{1:n})(\widehat{\theta} - \theta^*) = -\lambda_n \widehat{Z} + W^n + R^n$$

where $W^n := -\nabla \ell(\theta^*; X^{1:n})$ is the sample score function (that we will show is small with high probability), and R^n is the remainder term after coordinate-wise applications of the mean value theorem; $R^n_j = [\nabla^2 \ell(\theta^*; X^{1:n}) - \nabla^2 \ell(\bar{\theta}^{(j)}; X^{1:n})]^T_j (\widehat{\theta} - \theta^*)$, for some $\bar{\theta}^{(j)}$ on the line between $\widehat{\theta}$ and θ^*, and with $[\cdot]^T_j$ being the j-th row of a matrix.

Recalling the notation for the Fisher information matrix $Q^* := \nabla^2 \ell(\theta^*; X^{1:n})$, we then have

$$Q^*(\widehat{\theta} - \theta^*) = -\lambda_n \widehat{Z} + W^n + R^n.$$

From now on, we provide lemmas that respectively control various terms in the above expression: the score term W^n, the deviation $\widehat{\theta} - \theta^*$, and the remainder term R^n. The first lemma controls the score term W^n:

Lemma 9 *Suppose that we set* λ_n *to satisfy* $\frac{8(2-\alpha)}{\alpha}\sqrt{\kappa_1(n,p)\kappa_4}\sqrt{\frac{\log p}{n}} \leq \lambda_n \leq \frac{4(2-\alpha)}{\alpha}$ $\kappa_1(n,p)\kappa_2(n,p)\kappa_4$ *for some constant* $\kappa_4 \leq \min\{2\kappa_v/3, 2\kappa_h + \kappa_v\}$. *Suppose also that* $n \geq \frac{8\kappa_h^2}{\kappa_4^2}\log p$. *Then, given a incoherence parameter* $\alpha \in (0,1]$,

$$P\left(\frac{2-\alpha}{\lambda_n}\|W^n\|_\infty \leq \frac{\alpha}{4}\right) \geq 1 - c_1 p'^{-2} - \exp(-c_2 n) - \exp(-c_3 n)$$

where $p' := \max\{n, p\}$.

The next lemma controls the deviation $\widehat{\theta} - \theta^*$.

Lemma 10 *Suppose that* $\lambda_n d \leq \frac{\rho^2_{\min}}{40\rho_{\max}\kappa_3(n,p)\log p'}$ *and* $\|W^n\|_\infty \leq \frac{\lambda_n}{4}$. *Then, we have*

$$P\left(\|\widehat{\theta}_S - \theta^*_S\|_2 \leq \frac{5}{\rho_{\min}}\sqrt{d}\lambda_n\right) \geq 1 - c_1 p'^{-2}, \tag{28}$$

for some constant $c_1 > 0$.

The last lemma controls the Taylor series remainder term R^n:

Lemma 11 *If* $\lambda_n d \leq \frac{\rho^2_{\min}}{400\rho_{\max}\kappa_3(n,p)\log p'}\frac{\alpha}{2-\alpha}$, *and* $\|W^n\|_\infty \leq \frac{\lambda_n}{4}$, *then we have*

$$P\left(\frac{\|R^n\|_\infty}{\lambda_n} \leq \frac{\alpha}{4(2-\alpha)}\right) \geq 1 - c_1 p'^{-2}, \tag{29}$$

for some constant $c_1 > 0$.

The proof then follows from Lemmas 9, 10 and 11 in a straightforward fashion, following Ravikumar et al. (2010). Consider the choice of regularization parameter $\lambda_n = \frac{8(2-\alpha)}{\alpha}\sqrt{\kappa_1(n,p)\kappa_4}\sqrt{\frac{\log p}{n}}$. For a sample size greater $n \geq \max\{\frac{4}{\kappa_1(n,p)\kappa_2(n,p)^2\kappa_4}, \frac{8\kappa_h^2}{\kappa_4^2}\}\log p$, the conditions of Lemma 9 are satisfied, so that we may conclude that $\|W^n\|_\infty \leq \frac{\alpha}{1-\alpha}\frac{\lambda_n}{4} \leq \frac{\lambda_n}{4}$ with high probability. Moreover, with a sufficiently large sample size such that $n \geq L'\left(\frac{2-\alpha}{\alpha}\right)^4 d^2\kappa_1(n,p)\kappa_3(n,p)^2\log p(\log p')^2$ for some constant $L' > 0$ depending only on $\rho_{\min}$, $\rho_{\max}$, κ_4 and α, it can be shown that the remaining condition of Lemma 11 (and hence the milder condition in Lemma 10) in turn is satisfied. Therefore, the resulting statements (28) and (29) of Lemmas 10 and 11 hold with high probability.

Strict dual feasibility. Following Ravikumar et al. (2010), we obtain

$$\begin{aligned}\|\widehat{Z}_{S^c}\|_\infty &\leq |\!|\!|Q^*_{S^cS}(Q^*_{SS})^{-1}|\!|\!|_\infty\Big[\frac{\|W^n_S\|_\infty}{\lambda_n} + \frac{\|R^n_S\|_\infty}{\lambda_n} + 1\Big] + \frac{\|W^n_{S^c}\|_\infty}{\lambda_n} + \frac{\|R^n_{S^c}\|_\infty}{\lambda_n} \\ &\leq (1-\alpha) + (2-\alpha)\Big[\frac{\|W^n\|_\infty}{\lambda_n} + \frac{\|R^n\|_\infty}{\lambda_n}\Big] \leq (1-\alpha) + \frac{\alpha}{4} + \frac{\alpha}{4} = 1 - \frac{\alpha}{2} < 1.\end{aligned}$$

Correct sign recovery. To guarantee that the support of $\widehat{\theta}$ is not strictly within the true support S, it suffices to show that $\|\widehat{\theta}_S - \theta^*_S\|_\infty \leq \frac{\theta^*_{\min}}{2}$. From Lemma 10, we have $\|\widehat{\theta}_S - \theta^*_S\|_\infty \leq \|\widehat{\theta}_S - \theta^*_S\|_2 \leq \frac{5}{\rho_{\min}}\sqrt{d}\lambda_n \leq \frac{\theta^*_{\min}}{2}$ as long as $\theta^*_{\min} \geq \frac{10}{\rho_{\min}}\sqrt{d}\lambda_n$. This completes the proof.

D.1 Proof of Lemma 9

For a fixed $t \in V\backslash r$, we define $V_t^{(i)}$ for notational convenience so that

$$W^n_t = \frac{1}{n}\sum_{i=1}^n X^{(i)}_r X^{(i)}_t - X^{(i)}_t D'(\theta^*_r + \langle \theta^*_{\backslash r}, X^{(i)}_{V\backslash r}\rangle) = \frac{1}{n}\sum_{i=1}^n V_t^{(i)}$$

Consider the upper bound on the moment generating function of $V_t^{(i)}$, conditioned on $X^{(i)}_{V\backslash r}$,

$$\begin{aligned}&\mathbb{E}[\exp(aV_t)|X^{(i)}_{V\backslash r}] = \int_{X_r}\exp\Big\{a\big[X_rX^{(i)}_t - X^{(i)}_t D'\big(\theta^*_r + \langle\theta^*_{\backslash r}, X^{(i)}_{V\backslash r}\rangle\big)\big] \\ &\quad + \Big(X_r\big(\theta^*_r + \langle\theta^*_{\backslash r}, X^{(i)}_{V\backslash r}\rangle\big) + C(X_r) - D\big(\theta^*_r + \langle\theta^*_{\backslash r}, X^{(i)}_{V\backslash r}\rangle\big)\Big)\Big\} \\ &= \exp\Big\{D\big(\theta^*_r + \langle\theta^*_{\backslash r}, X^{(i)}_{V\backslash r}\rangle + aX^{(i)}_t\big) - D\big(\theta^*_r + \langle\theta^*_{\backslash r}, X^{(i)}_{V\backslash r}\rangle\big) - aX^{(i)}_t D'\big(\theta^*_r + \langle\theta^*_{\backslash r}, X^{(i)}_{V\backslash r}\rangle\big)\Big\} \\ &= \exp\Big\{\frac{a^2}{2}\big(X^{(i)}_t\big)^2 D''\big(\theta^*_r + \langle\theta^*_{\backslash r}, X^{(i)}_{V\backslash r}\rangle + \nu_i aX^{(i)}_t\big)\Big\} \quad \text{for some } \nu_i \in [0,1]\end{aligned}$$

where the last equality holds by the second-order Taylor series expansion. Consequently, we have

$$\frac{1}{n}\sum_{i=1}^n \log\mathbb{E}[\exp(aV^{(i)}_t)|X^{(i)}_{V\backslash r}] \leq \frac{1}{n}\sum_{i=1}^n \frac{a}{2}\big(X^{(i)}_t\big)^2 D''\big(\theta^*_r + \langle\theta^*_{\backslash r}, X^{(i)}_{V\backslash r}\rangle + \nu_i aX^{(i)}_t\big).$$

First, we define the event: $\xi_1 := \Big\{ \max_{i,s} |X_r^{(i)}| \leq 4 \log p' \Big\}$. Then, by Proposition 4, we obtain $P[\xi_1^c] \leq c_1\, npp'^{-4} \leq c_1\, p'^{-2}$. Provided that $a \leq \kappa_2(n,p)$, we can use Condition (C4) to control the second-order derivative of log-partition function and we obtain

$$\frac{1}{n}\sum_{i=1}^{n} \log \mathbb{E}[\exp(aV_t^{(i)})|X_{V\setminus r}^{(i)}] \leq \frac{\kappa_1(n,p)a^2}{2}\frac{1}{n}\sum_{i=1}^{n}\big(X_t^{(i)}\big)^2 \quad \text{for } a \leq \kappa_2(n,p)$$

with probability at least $1 - c_1 p'^{-2}$. Now, for each index t, the variables $\frac{1}{n}\sum_{i=1}^{n}\big(X_t^{(i)}\big)^2$ satisfy the tail bound in Proposition 3. Let us define the event $\xi_2 := \Big\{ \max_{t\in V} \frac{1}{n}\sum_{i=1}^{n}\big(X_t^{(i)}\big)^2 \leq \kappa_4 \Big\}$ for some constant $\kappa_4 \leq \min\{2\kappa_v/3, 2\kappa_h + \kappa_v\}$. Then, we can establish the upper bound of probability $P[\xi_2^c]$ by a union bound:

$$P[\xi_2^c] \leq \exp(-\frac{\kappa_4^2}{4\kappa_h^2}n + \log p) \leq \exp(-c_2 n)$$

as long as $n \geq \frac{8\kappa_h^2}{\kappa_4^2}\log p$. Therefore, conditioned on ξ_1, ξ_2, the moment generating function is bounded as follows:

$$\frac{1}{n}\sum_{i=1}^{n} \log \mathbb{E}[\exp(aV_t^{(i)})|X_{V\setminus r}^{(i)}, \xi_1, \xi_2] \leq \frac{\kappa_1(n,p)\kappa_4\, a^2}{2} \quad \text{for } a \leq \kappa_2(n,p).$$

The standard Chernoff bound technique implies that for any $\delta > 0$,

$$P\Big[\big|\frac{1}{n}\sum_{i=1}^{n} V_t^{(i)}\big| > \delta \mid \xi_1, \xi_2\Big] \leq 2\exp\left(n\Big(\frac{\kappa_1(n,p)\kappa_4\, a^2}{2} - a\delta\Big)\right) \quad \text{for } a \leq \kappa_2(n,p).$$

Setting $a = \frac{\delta}{\kappa_1(n,p)\kappa_4}$ yields

$$P\Big[\big|\frac{1}{n}\sum_{i=1}^{n} V_t^{(i)}\big| > \delta \mid \xi_1, \xi_2\Big] \leq 2\exp\left(-\frac{n\delta^2}{2\kappa_1(n,p)\kappa_4}\right) \quad \text{for } \delta \leq \kappa_1(n,p)\kappa_2(n,p)\kappa_4.$$

Suppose that $\frac{\alpha}{2-\alpha}\frac{\lambda_n}{4} \leq \kappa_1(n,p)\kappa_2(n,p)\kappa_4$ for large enough n; thus setting $\delta = \frac{\alpha}{2-\alpha}\frac{\lambda_n}{4}$:

$$P\Big[\big|\frac{1}{n}\sum_{i=1}^{n} V_t^{(i)}\big| > \frac{\alpha}{2-\alpha}\frac{\lambda_n}{4} \mid \xi_1, \xi_2\Big] \leq 2\exp\left(-\frac{\alpha^2}{(2-\alpha)^2}\frac{n\lambda_n^2}{32\kappa_1(n,p)\kappa_4}\right),$$

and by a union bound, we obtain

$$P\Big[\|W^n\|_\infty > \frac{\alpha}{2-\alpha}\frac{\lambda_n}{4} \mid \xi_1, \xi_2\Big] \leq 2\exp\left(-\frac{\alpha^2}{(2-\alpha)^2}\frac{n\lambda_n^2}{32\kappa_1(n,p)\kappa_4} + \log p\right).$$

Finally, provided that $\lambda_n \geq \frac{8(2-\alpha)}{\alpha}\sqrt{\kappa_1(n,p)\kappa_4}\sqrt{\frac{\log p}{n}}$, we obtain

$$P\Big[\|W^n\|_\infty > \frac{\alpha}{2-\alpha}\frac{\lambda_n}{4}\Big] \leq c_1 p'^{-2} + \exp(-c_2 n) + \exp(-c_3 n)$$

where we use the fact that the probability of occurring event $\mathcal{A}$ is upper bounded by $P(\mathcal{A}) \leq P(\xi_1^c) + P(\xi_2^c) + P(\mathcal{A}|\xi_1, \xi_2)$.

D.2 Proof of Lemma 10

In order to establish the error bound $\|\widehat{\theta}_S - \theta^*_S\|_2 \leq B$ for some radius B, several works (e.g. Negahban et al. (2012); Ravikumar et al. (2010)) proved that it suffices to show $F(u_S) > 0$ for all $u_S := \theta_S - \theta^*_S$ s.t. $\|u_S\|_2 = B$ where

$$F(u_S) := \ell(\theta^*_S + u_S; X^{1:n}) - \ell(\theta^*_S; X^{1:n}) + \lambda_n(\|\theta^*_S + u_S\|_1 - \|\theta^*_S\|_1).$$

Note that $F(0) = 0$, and for $\hat{u}_S := \widehat{\theta}_S - \theta^*_S$, $F(\hat{u}_S) \leq 0$. From now on, we show that $F(u_S)$ is strictly positive on the boundary of the ball with radius $B = M\lambda_n\sqrt{d}$ where $M > 0$ is a parameter that we will choose later in this proof. Some algebra yields

$$F(u_S) \geq (\lambda_n\sqrt{d})^2\Big\{ -\frac{1}{4}M + q^*M^2 - M\Big\} \tag{30}$$

where q^* is the minimum eigenvalue of $\nabla^2\ell(\theta^*_S + vu_S; X^{1:n})$ for some $v \in [0,1]$. Moreover,

$$\begin{aligned}
q^* &:= \Lambda_{\min}\big(\nabla^2\ell(\theta^*_S + vu_S)\big) \\
&\geq \min_{v\in[0,1]} \Lambda_{\min}\big(\nabla^2\ell(\theta^*_S + vu_S)\big) \\
&\geq \Lambda_{\min}\big[\frac{1}{n}\sum_{i=1}^n D''(\theta^*_r + \langle\theta^*_S, X^{(i)}_S\rangle)X^{(i)}_S\big(X^{(i)}_S\big)^T\big] \\
&\qquad - \max_{v\in[0,1]} \Big\|\frac{1}{n}\sum_{i=1}^n D'''\big(\theta^*_r + \langle\theta^*_S, X^{(i)}_S\rangle\big)\big(u_S^TX^{(i)}_S\big)X^{(i)}_S\big(X^{(i)}_S\big)^T\Big\|_2 \\
&\geq \rho_{\min} - \max_{v\in[0,1]}\max_y \frac{1}{n}\sum_{i=1}^n |D'''\big(\theta^*_r + \langle\theta^*_S, X^{(i)}_S\rangle\big)|\ |\langle u_S, X^{(i)}_S\rangle|\ \big(\langle X^{(i)}_S, y\rangle\big)^2
\end{aligned}$$

where $y \in \mathbb{R}^d$ s.t $\|y\|_2 = 1$. Similarly as in the previous proof, we consider the event ξ_1 with probability at least $1 - c_1p'^{-2}$. Then, since all the elements in vector $X^{(i)}_S$ is smaller than $4\log p'$, $|\langle u_S, X^{(i)}_S\rangle| \leq 4\log p'\sqrt{d}\|u_S\|_2 = 4\log p'M\lambda_nd$ for all i. At the same time, by Condition (C4), $|D'''((\theta^*_r + vu_S) + \langle\theta^*_S + vu_S, X^{(i)}_S\rangle)| \leq \kappa_3(n,p)$. Note that $\theta^*_S + vu_S$ is a convex combination of θ^*_S and $\widehat{\theta}_S$, and as a result, we can directly apply the Condition (C4). Hence, conditioned on ξ_1, we have

$$q^* \geq \rho_{\min} - 4\rho_{\max}M\lambda_nd\kappa_3(n,p)\log p'.$$

As a result, assuming that $\lambda_n \leq \frac{\rho_{\min}}{8\rho_{\max}Md\kappa_3(n,p)\log p'}$, $q^* \geq \frac{\rho_{\min}}{2}$. Finally, from (30), we obtain

$$F(u_S) \geq (\lambda_n\sqrt{d})^2\Big\{ -\frac{1}{4}M + \frac{\rho_{\min}}{2}M^2 - M\Big\},$$

which is strictly positive for $M = \frac{5}{\rho_{\min}}$. Therefore, if $\lambda_nd \leq \frac{\rho_{\min}}{8\rho_{\max}M\kappa_3(n,p)\log p'}$ $\leq \frac{\rho^2_{\min}}{40\rho_{\max}\kappa_3(n,p)\log p'}$, then

$$\|\widehat{\theta}_S - \theta^*_S\|_2 \leq \frac{5}{\rho_{\min}}\sqrt{d}\lambda_n,$$

which completes the proof.

D.3 Proof of Lemma 11

In the proof, we are going to show that $\|R^n\|_\infty \leq 4\kappa_3(n,p)\log p'\rho_{\max}\|\widehat{\theta}_S - \theta^*_S\|_2^2$. Then, since the conditions of Lemma 11 are stronger than those of Lemma 10, from the result of Lemma 10, we can conclude that

$$\|R^n\|_\infty \leq \frac{100\kappa_3(n,p)\rho_{\max}\log p'}{\rho^2_{\min}}\lambda_n^2 d,$$

as claimed in Lemma 11.

From the definition of R^n, for a fixed $t \in V\backslash r$, R^n_t can be written as

$$\frac{1}{n}\sum_{i=1}^n \Big[D''\big(\theta^*_r + \langle\theta^*_{\backslash r}, X^{(i)}_{V\backslash r}\rangle\big) - D''\big(\bar{\theta}_s + \langle\bar{\theta}^{(t)}, X^{(i)}_{V\backslash r}\rangle\big)\Big]\Big[X^{(i)}_{V\backslash r}\big(X^{(i)}_{V\backslash r}\big)^T\Big]^T_t[\widehat{\theta}_{\backslash r} - \theta^*_{\backslash r}]$$

where $\bar{\theta}^{(t)}_{\backslash r}$ is some point in the line between $\widehat{\theta}_{\backslash r}$ and $\theta^*_{\backslash r}$, i.e., $\bar{\theta}^{(t)}_{\backslash r} = v_t\widehat{\theta}_{\backslash r} + (1-v_t)\theta^*_{\backslash r}$ for $v_t \in [0,1]$. By another application of the mean value theorem, we have

$$R^n_t = -\frac{1}{n}\sum_{i=1}^n \Big\{D'''\big(\bar{\bar{\theta}}_s + \langle\bar{\bar{\theta}}^{(t)}_{\backslash r}, X^{(i)}_{V\backslash r}\rangle\big)X^{(i)}_t\Big\}\Big\{v_t[\widehat{\theta}_{\backslash r} - \theta^*_{\backslash r}]^T X^{(i)}_{V\backslash r}\big(X^{(i)}_{V\backslash r}\big)^T[\widehat{\theta}_{\backslash r} - \theta^*_{\backslash r}]\Big\}$$

for a some point $\bar{\bar{\theta}}^{(t)}_{\backslash r}$ between $\bar{\theta}^{(t)}_{\backslash r}$ and $\theta^*_{\backslash r}$. Similarly in the previous proofs, conditioned on the event ξ_1, we obtain

$$|R^n_t| \leq \frac{4\kappa_3(n,p)\log p'}{n}\sum_{i=1}^n \Big\{v_t[\widehat{\theta}_{\backslash r} - \theta^*_{\backslash r}]^T X^{(i)}_{V\backslash r}\big(X^{(i)}_{V\backslash r}\big)^T[\widehat{\theta}_{\backslash r} - \theta^*_{\backslash r}]\Big\}.$$

Performing some algebra yields

$$|R^n_t| \leq 4\kappa_3(n,p)\rho_{\max}\log p'\,\|\widehat{\theta}_S - \theta^*_S\|_2^2, \quad \text{for all } t \in V\backslash r$$

with probability at least $1 - c_1 p'^{-2}$, which completes the proof.

Appendix E. Optimization Problems for Poisson and Exponential Graphical Model Neighborhood Selection

We propose to fit our family of graphical models by performing neighborhood selection, or maximizing the ℓ_1-penalized log-likelihood for each node conditional on all other nodes. For several exponential families, however, further restrictions on the parameter space are needed to ensure a proper Markov Random Field. When performing neighborhood selection, these can be imposed by adding constraints to the penalized generalized linear models. We illustrate this by providing the optimization problems solved by our Poisson graphical model and exponential graphical model M-estimator that are used in Section 4.

Following from Section 3, the neighborhood selection problem for our family of models maximizes the likelihood of a node, X_r, conditional on all other nodes, $X_{V\backslash r}$. This conditional likelihood is regularized with an ℓ_1 penalty to induce sparsity in the edge weights,

$\theta(r)$ a $p-1$ dimensional vector, and constrained to enforce restrictions, $\theta(r) \in \mathcal{C}$, needed to yield a proper MRF:

$$\underset{\theta(r)}{\text{maximize}} \quad \frac{1}{n}\sum_{i=1}^{n} \ell\left(X_{i,r}|X_{i,V\setminus r};\theta(r)\right) - \lambda_n\|\theta(r)\|_1 \;\; \text{subject to} \;\; \theta(r) \in \mathcal{C},$$

where $\ell\left(X_{i,r}|X_{i,V\setminus r};\theta(r)\right)$ is the conditional log-likelihood for the exponential family. For the Poisson graphical model, the edge weights are constrained to be non-positive. This yields the following optimization problem:

$$\begin{aligned} &\underset{\theta(r)}{\text{maximize}} \quad \frac{1}{n}\sum_{i=1}^{n}\left[X_{r,i}X_{V\setminus r,i}^T\theta(r) - \exp\left(X_{V\setminus r,i}^T\theta(r)\right)\right] - \lambda_n\|\theta(r)\|_1 \\ &\text{subject to} \quad \theta(r) \leq 0. \end{aligned}$$

Similarly, the edge weights of the exponential graphical are restricted to be non-negative yielding

$$\begin{aligned} &\underset{\theta(r)}{\text{maximize}} \quad \frac{1}{n}\sum_{i=1}^{n}\left[-X_{r,i}X_{V\setminus r,i}^T\theta(r) + \log\left(X_{V\setminus r,i}^T\theta(r)\right)\right] - \lambda_n\|\theta(r)\|_1 \\ &\text{subject to} \quad \theta(r) \geq 0. \end{aligned}$$

Note that we neglect the intercept term, assuming this to be zero as is common in other proposed neighborhood selection methods (Meinshausen and Bühlmann, 2006; Ravikumar et al., 2010; Jalali et al., 2011). Both of the neighborhood selection problems are concave problems with a smooth log-likelihood and linear constraints. While there are a plethora of optimization routines available to solve such problems, we have employed a projected gradient descent scheme which is guaranteed to converge to a global optimum (Daubechies et al., 2008; Beck and Teboulle, 2010).

References

K. Abdelmohsen, M. M. Kim, S. Srikantan, E. M. Mercken, S. E. Brennan, G. M. Wilson, R. de Cabo, and M. Gorospe. mir-519 suppresses tumor growth by reducing hur levels. *Cell Cycle*, 9(7):1354–1359, 2010.

D. Acemoglu. The crisis of 2008: Lessons for and from economics. *Critical Review*, 21(2-3): 185–194, 2009.

G. I. Allen and Z. Liu. A local poisson graphical model for inferring networks from sequencing data. *IEEE Trans. NanoBioscience*, 12(1):1–10, 2013.

A. Beck and M. Teboulle. Gradient-based algorithms with applications to signal recovery problems. *Convex Optimization in Signal Processing and Communications*, pages 42–88, 2010.

J. Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, 36(2):192–236, 1974.

Y. M. Bishop, S. E. Fienberg, and P. W. Holland. *Discrete Multivariate Analysis*. Springer Verlag, 2007.

P. Bühlmann. *Statistics for High-dimensional Data*. Springerverlag Berlin Heidelberg, 2011.

J. Bullard, E. Purdom, K. Hansen, and S. Dudoit. Evaluation of statistical methods for normalization and differential expression in mrna-seq experiments. *BMC Bioinformatics*, 11(1):94, 2010.

Cancer Genome Atlas Research Network. Comprehensive molecular portraits of human breast tumours. *Nature*, 490(7418):61–70, 2012.

P. Clifford. Markov random fields in statistics. In *Disorder in Physical Systems*. Oxford Science Publications, 1990.

I. Daubechies, M. Fornasier, and I. Loris. Accelerated projected gradient method for linear inverse problems with sparsity constraints. *Journal of Fourier Analysis and Applications*, 14(5):764–792, 2008.

A. Dobra and A. Lenkoski. Copula gaussian graphical models and their application to modeling functional disability data. *Annals of Applied Statistics*, 5(2A):969–993, 2011.

J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the lasso. *Biostatistics*, 9(3):432–441, 2007.

N. Friedman. Inferring cellular networks using probabilistic graphical models. *Science*, 303 (5659):799–805, 2004.

L. A. Herzenberg, J. Tung, W. A. Moore, L. A. Herzenberg, and D. R. Parks. Interpreting flow cytometry data: a guide for the perplexed. *Nature Immunology*, 7(7):681–685, 2006.

A. Jalali, P. Ravikumar, V. Vasuki, and S. Sanghavi. On learning discrete graphical models using group-sparse regularization. In *Inter. Conf. on AI and Statistics (AISTATS)*, 14, 2011.

I. Keklikoglou, C. Koerner, C. Schmidt, J. D. Zhang, D. Heckmann, A. Shavinskaya, H. Allgayer, B. Gückel, T. Fehm, A. Schneeweiss, et al. Microrna-520/373 family functions as a tumor suppressor in estrogen receptor negative breast cancer by targeting nf-κb and tgf-β signaling pathways. *Oncogene*, 31:4150–4163, 2012.

J. Lafferty, H. Liu, and L. Wasserman. Sparse nonparametric graphical models. *Statistical Science*, 27(4):519–537, 2012.

S. L. Lauritzen. *Graphical Models*. Oxford University Press, USA, 1996.

J. T. Leek, R. B. Scharpf, H. C. Bravo, D. Simcha, B. Langmead, W. E. Johnson, D. Geman, K. Baggerly, and R. A. Irizarry. Tackling the widespread and critical impact of batch effects in high-throughput data. *Nature Reviews Genetics*, 11(10):733–739, 2010.

J. Li, D. M. Witten, I. M. Johnstone, and R. Tibshirani. Normalization, testing, and false discovery rate estimation for rna-sequencing data. *Biostatistics*, 2011.

H. Liu, J. Lafferty, and L. Wasserman. The nonparanormal: Semiparametric estimation of high dimensional undirected graphs. *Journal of Machine Learning Research*, 10:2295–2328, 2009.

H. Liu, K. Roeder, and L. Wasserman. Stability approach to regularization selection (stars) for high dimensional graphical models. *Arxiv preprint arXiv:1006.3316*, 2010.

H. Liu, F. Han, M. Yuan, J. Lafferty, and L. Wasserman. High dimensional semiparametric gaussian copula graphical models. *Arxiv preprint arXiv:1202.2169*, 2012a.

H. Liu, F. Han, M. Yuan, J. Lafferty, and L. Wasserman. The nonparanormal skeptic. In *International Conference on Machine learning (ICML)*, 29, 2012b.

R. Lockhart, J. Taylor, R. J. Tibshirani, and R. Tibshirani. A significance test for the lasso. *Annals of Statistics*, 42(2):413–468, 2014.

J. C. Marioni, C. E. Mason, S. M. Mane, M. Stephens, and Y. Gilad. Rna-seq: an assessment of technical reproducibility and comparison with gene expression arrays. *Genome Research*, 18(9):1509–1517, 2008.

P. McCullagh and J. A. Nelder. *Generalized Linear Models*. Monographs on statistics and applied probability 37. Chapman and Hall/CRC, 1989.

N. Meinshausen and P. Bühlmann. High-dimensional graphs and variable selection with the Lasso. *Annals of Statistics*, 34:1436–1462, 2006.

A. Mortazavi, B. A. Williams, K. McCue, L. Schaeffer, and B. Wold. Mapping and quantifying mammalian transcriptomes by rna-seq. *Nature Methods*, 5(7):621–628, 2008.

S. Negahban, P. Ravikumar, M. J. Wainwright, and B. Yu. A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers. *Statistical Science*, 27:538–557, 2012.

D. Pe'er, A. Regev, G. Elidan, and N. Friedman. Inferring subnetworks from perturbed expression profiles. *Bioinformatics*, 17(Suppl 1):S215–S224, June 2001.

P. Ravikumar, M. J. Wainwright, and J. Lafferty. High-dimensional ising model selection using ℓ_1-regularized logistic regression. *Annals of Statistics*, 38(3):1287–1319, 2010.

K. Sachs, O. Perez, D. Pe'er, D. A. Lauffenburger, and G.P. Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308(5721):523–529, 2005.

T. P. Speed and H. T. Kiiveri. Gaussian Markov distributions over finite graphs. *Annals of Statistics*, 14(1):138–150, 1986.

C. Varin and P. Vidoni. A note on composite likelihood inference and model selection. *Biometrika*, 92:519–528, 2005.

C. Varin, N. Reid, and D. Firth. An overview of composite likelihood methods. *Statistica Sinica*, 21:5–42, 2011.

M. J. Wainwright. Sharp thresholds for high-dimensional and noisy sparsity recovery using ℓ_1-constrained quadratic programming (Lasso). *IEEE Trans. Information Theory*, 55: 2183–2202, 2009.

M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1-2):1–305, 2008.

Z. Wei and H. Li. A markov random field model for network-based analysis of genomic data. *Bioinformatics*, 23(12):1537–1544, 2007.

L. Xue and H. Zou. Regularized rank-based estimation of high-dimensional nonparanormal graphical models. *Annals of Statistics*, 40:2541–2571, 2012.

E. Yang, P. Ravikumar, G. I. Allen, and Z. Liu. Graphical models via generalized linear models. In *Neur. Info. Proc. Sys. (NIPS)*, 25, 2012.

F. Yu, H. Yao, P. Zhu, X. Zhang, Q. Pan, C. Gong, Y. Huang, X. Hu, F. Su, J. Lieberman, et al. let-7 regulates self renewal and tumorigenicity of breast cancer cells. *Cell*, 131(6): 1109–1123, 2007.

Journal of Machine Learning Research 16 (2015) 3849-3875 Submitted 9/14; Published 12/15

Marginalizing Stacked Linear Denoising Autoencoders

Minmin Chen M.CHEN@CRITEO.COM
Criteo
Palo Alto, CA 94301, USA

Kilian Q. Weinberger KILIAN@WUSTL.EDU
Zhixiang (Eddie) Xu XUZX@CSE.WUSTL.EDU
Department of Computer Science and Engineering
Washington University in St. Louis
St. Louis, MO 63130, USA

Fei Sha FEISHA@USC.EDU
Computer Science Department
University of Southern California
Los Angeles, CA 90089, USA

Editor: Leon Bottou

Abstract

Stacked denoising autoencoders (SDAs) have been successfully used to learn new representations for domain adaptation. They have attained record accuracy on standard benchmark tasks of sentiment analysis across different text domains. SDAs learn robust data representations by reconstruction, recovering original features from data that are artificially corrupted with noise. In this paper, we propose *marginalized Stacked Linear Denoising Autoencoder* (mSLDA) that addresses two crucial limitations of SDAs: high computational cost and lack of scalability to high-dimensional features. In contrast to SDAs, our approach of mSLDA marginalizes noise and thus does not require stochastic gradient descent or other optimization algorithms to learn parameters — in fact, the linear formulation gives rise to a closed-form solution. Consequently, mSLDA, which can be implemented in only 20 lines of MATLAB$^{\mathrm{TM}}$, is about *two orders of magnitude* faster than a corresponding SDA. Furthermore, the representations learnt by mSLDA are as effective as the traditional SDAs, attaining almost identical accuracies in benchmark tasks.

Keywords: domain adaption, fast representation learning, noise marginalization, denoising autoencoders

1. Introduction

The goal of domain adaptation (Ben-David et al., 2010; Huang et al., 2006; Weinberger et al., 2009; Xue et al., 2008) is to generalize a classifier that is trained on a source domain, for which typically plenty of training data is available, to a target domain, for which data is scarce. Cross-domain generalization is important in many application areas of machine learning, where such an imbalance of training data may occur. Examples include computational biology (Liu et al., 2008), natural language processing (Daume III, 2007; McClosky et al., 2006), computer vision (Saenko et al., 2010) and web-search ranking (Chapelle et al., 2011).

Adaptation is challenging, because the data in the two domains is not identically distributed and a classifier trained on source can be expected to perform significantly worse on the target domain. Recent work has investigated several techniques to reduce this adaptation error:

- *instance re-weighting* (Huang et al., 2006; Mansour et al., 2009) is an approach to re-weight source inputs so that the distribution of the reweighed source data matches that of the target domain; instance weighting strategies assume that the source and target distribution share the same support and features. It tends to be less effective for tasks of high-dimensional, sparse features such as text documents and where source and target distributions differ more drastically.

- *joint feature mapping* (Blitzer et al., 2006; Gong et al., 2012; Xue et al., 2008; Glorot et al., 2011) is an approach to learn a new shared representation for the source and target domains, in which the two data distributions align. These algorithms are designed for highly divergent domains, which can contain different features, and are more closely related to our work.

- *parameter sharing* (Daume III, 2007; Chapelle et al., 2011; Weinberger et al., 2009) is an approach to adapt machine learning classifiers to incorporate shared weights across the two domains. This is arguably the most popular category of domain adaptation algorithms amongst practitioners, mostly due to their appealing simplicity (Daume III, 2007).

One of the most successful domain adaptation algorithms was introduced by Glorot et al. (2011), which falls into the second category. The authors use stacked denoising autoencoders (SDA) (Vincent et al., 2008) to learn a joint feature representation that can be shared across multiple domains. Denoising autoencoders are one-layer neural networks that are optimized to reconstruct input data from partial and random corruption. These denoisers can be stacked into deep learning architectures, which are then fine-tuned with back-propagation (Vincent et al., 2008). Glorot et al. (2011) use the internal representation of the intermediate layers of the SDA as input features for linear classifiers, an idea pioneered by Lee et al. (2009) and Vincent et al. (2010). The authors demonstrate in their work that such SDA-learned features are very effective for cross-domain generalization, even with straight-forward linear Support Vector Machines (SVM) (Cortes and Vapnik, 1995). For example, it yields record adaptation accuracies on the Amazon$^{\text{TM}}$ sentiment-analysis benchmark tasks of predicting review sentiment across product domains (Blitzer et al., 2006).

Although the capabilities of SDAs are remarkable, they are limited by their high computational cost. Compared with competing approaches (Blitzer et al., 2006; Xue et al., 2008; Chen et al., 2011b), SDAs are significantly slower to train. This is primarily the case because of the large number of model parameters in the denoising autoencoders, which are learned with iterative algorithms for numerical optimization. The challenge is further compounded by the dimensionality of the input data and the need for computationally intensive model selection procedures to tune hyperparameters. Consequently, even a highly optimized implementation (Bergstra et al., 2010) may require hours (even days) of training time on the larger Amazon$^{\text{TM}}$ benchmark data sets.

In this paper, we introduce a variation of SDAs that addresses these shortcomings. The proposed method, which we refer to as *marginalized Stacked Linear Denoising Autoencoder* (mSLDA), adopts the greedy layer-by-layer training of SDAs. Similarly, at each layer we learn a denoiser to recover input data from random corruption. However, a crucial difference is that we use *linear* denoisers as the basic building blocks. This restriction has two important advantages: 1. the random

feature corruption can be marginalized out, which alleviates the need to iterate over many corrupted versions of the data; 2. the weights of the linear denoisers can be computed in closed form, in very little time (almost instantaneous). Conceptually, marginalizing the corruption is equivalent to training the model over an infinite number of corrupted versions of the input data.

Although the restriction to only linear denoisers makes mSLDA less expressive than SDA, we observe that for high dimensional data sets they are sufficient and mSLDA features match the original SDA features in quality. This is particularly impressive, as the training of the mSLDA features is several orders of magnitude faster (reducing training from up to 2 days for SDA to a few minutes with mSLDA).

Two earlier short paper on this work (Chen et al., 2012; Xu et al., 2012), already introduce this learning framework, but this longer version provides a significant amount of additional details. In particular, we provide extensions to different corruption models, further and deeper analysis of the mSLDA algorithm, additional experiments with different datasets (text documents and images), and new experimental results in semi-supervised settings. The remaining parts of the paper is organized as follows. In Section 2 we lay out the problem and review a couple of closely-related prior works. In Section 3 we introduce the mSLDA framework for learning representations. In Section 4 we discuss several input corruption models, which fit naturally into the mSLDA framework. In Section 5 we propose an extension to scale up our learning framework to inputs of high dimensions. In Section 7 we present an extensive set of results evaluating mSLDA on several text classification and object recognition tasks. In Section 8 we provide further analysis of the results and discuss strengths and limitations of mSLDA.

2. Background and Related Work

We assume that during training we are provided with labeled data from the source domain $L = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\} \subset \mathcal{R}^d$ with corresponding labels $y_1, \ldots, y_m \subset \mathcal{Y}$. Here, $\mathcal{Y}$ can consist of real valued or categorical labels. We focus on the simple binary case with $\mathcal{Y} = \{+1, -1\}$ throughout this manuscript, however we would like to emphasize that our proposed feature learning algorithm is *unsupervised* and therefore agnostic to the label choice (which only affects the classifier trained on the learned features). If labeled *target* data is available, it can be included into L, although in our setting we do not assume this is the case. We are potentially also provided with *unlabeled* data $U = \{\mathbf{x}_{m+1} \ldots, \mathbf{x}_{m+u}\} \subset \mathcal{R}^d$, which may be sampled from source, target or other (related) source distributions. For notational simplicity we define $n = m + u$. Although we do assume that any two domains have some overlap in features, we *do not* assume that they have *identical* features. Instead, we pad all input vectors with zeros to make them of matching dimensionality d. Given this mix of labeled and unlabeled source and target data, our goal is to train a classifier that accurately predicts the labels of instances from the target domain T.

In the following, we briefly review work that is most similar to ours, including Structural Correspondence Learning (SCL) (Blitzer et al., 2006), Stacked Denoising Autoencoders (Glorot et al., 2011) and learning with marginalized corruption (van der Maaten et al., 2013).

2.1 Structural Correspondence Learning

The leaning of joint source / target representations explicitly for domain adaptation was pioneered by Blitzer et al. (2006) and their Structural Correspondence Learning (SCL) algorithm. SCL assumes a known set of pivot features, which appear frequently in both domains (source and target)

and behave similarly. These are used to put domain specific words in correspondence. The low-rank representation learned with SCL essentially encodes the covariance between non-pivot features and the pivot features. As described in detail in Section 3, the single-layer mSLDA also learns the correlations between *all* the features. In this sense, the resulting feature space is similar to SCL, and the computation time of SCL and mSLDA are comparable. However, mSLDA introduces reconstruction from corruption and stacking of multiple denoising layers, which result in superior feature quality. Further, mSLDA does not require any side information about a pivot features set, which can be hard to identify (Blitzer et al., 2006).

2.2 Marginalized Corrupted Features

Recently, van der Maaten et al. (2013) proposed the Marginalized Corrupted Features (MCF) learning framework, which was inspired by our earlier publication of mSLDA (Chen et al., 2012).[1] MCF uses marginalized corruption to improve the generalization performance of linear classifiers, as an alternative to L_2 or L_1 norm regularization. MCF is equivalent to first generating infinitely many corrupted copies of the training data, with a pre-defined corruption distribution, and then training an unregularized classifier on this (infinite) data set. Training on additional corrupted inputs leads to substantially more robust classifiers, as has previously been shown by Burges and Schölkopf (1997). MCF borrows the idea from mSLDA to marginalize out this corruption, which leads to substantial improvements in speed and accuracy over explicitly corrupting only finitely many copies of the training data. In a similar spirit, Wang and Manning (2013) introduce marginalized dropout (Hinton et al., 2012) for logistic regression and show that the marginalized corruption can be interpreted as active regularization.

2.3 Stacked Denoising Autoencoder

Our work is mostly inspired by Autoencoders. Various forms of autoencoders have been developed in the machine learning community (Rumelhart et al., 1986; Baldi and Hornik, 1989; Kavukcuoglu et al., 2009; Lee et al., 2009; Vincent et al., 2008; Rifai et al., 2011). In its simplest form, an autoencoder has two components, an encoder $h(\cdot)$ maps an input $\mathbf{x} \in \mathcal{R}^d$ to some hidden representation $h(\mathbf{x}) \in \mathcal{R}^{d_h}$, and a decoder $g(\cdot)$ maps this hidden representation back to a reconstructed version of $\mathbf{x}$, such that $g(h(\mathbf{x})) \approx \mathbf{x}$. The parameters of the autoencoders are learned to minimize the reconstruction error, measured by some loss $\ell(\mathbf{x}, g(h(\mathbf{x})))$. Choices for the loss include squared error or Kullback-Leibler divergence (when the feature values are in $[0, 1]$.)

Denoising Autoencoders (DAs) incorporate a slight modification to this setup and corrupt the inputs before mapping them into the hidden representation. They are trained to reconstruct (or *denoise*) the original input $\mathbf{x}$ from its corrupted version $\tilde{\mathbf{x}}$ by minimizing $\ell(\mathbf{x}, g(h(\tilde{\mathbf{x}})))$. Typical choices of corruption include additive isotropic Gaussian noise or binary masking noise. As in Vincent et al. (2008), we primarily use the latter and set a fraction of the features of each input to *zero*. This is a natural choice for bag-of-word representations of text documents, where author specific word preferences can influence the existence or absence of words in the source and target domains.

The stacked denoising autoencoder (SDA) of Vincent et al. (2008) stacks several DAs together to create higher-level representations, by feeding the hidden representation of the t^{th} DA as input into the $(t+1)^{th}$ DA. The training is performed greedily, layer by layer.

1. In this earlier work we refer to mSLDA as simply marginalized Stacked Denoising Autoencoder (mSDA). Since then we added the term "Linear" to avoid confusion.

Feature Generation. Recently, Lee et al. (2009) and Glorot et al. (2011) have identified autoencoders as a powerful tool for automatic discovery and extraction of nonlinear features. For example, Lee et al. (2009) demonstrate that the hidden representations computed by all or partial layers of a convolutional deep belief network (CDBN) make excellent features for classification with SVMs. The pre-processing with a CDBN improves the generalization by increasing robustness against noise and label-invariant transformations.

Glorot et al. (2011) successfully apply SDAs to extract features for domain adaptation in document sentiment analysis. The authors train an SDA to reconstruct the unlabeled input vectors on the union of the source and target data. A classifier (*e.g.* a linear SVM) trained on the resulting feature representation $h(\mathbf{x})$ transfers significantly better from source to target than one trained on $\mathbf{x}$ directly. Similar to CDBNs, SDAs also combine correlated input dimensions, as they reconstruct removed feature values from the remaining uncorrupted ones. In fact, Glorot et al. (2011) show that SDAs are able to disentangle hidden factors, which explain the variations in the input data, and automatically group features in accordance with their relatedness to these factors. This helps transfer across domains as these generic concepts are invariant to domain-specific vocabularies.

As an intuitive example, imagine that we classify product reviews according to their sentiments. The source data consists of *book* reviews, the target of *kitchen appliances*. A classifier trained on the original bag-of-words source never encounters the bigram *energy efficient* during training and therefore assigns zero weight to it. In the learned SDA representation, the bigram *energy efficient* would tend to reconstruct, and be reconstructed by, co-occurring features, typically of similar sentiment (*e.g. good* or *love*). The SDA will preform the same reconstruction also on the source data, in other words, it will "reconstruct" bigrams like *energy efficient* in book reviews that contain words with positive sentiment. Thus, the source-trained classifier can assign weights even to features that never occur in its original domain representation.

Although SDAs generate excellent features for domain adaptation, they have several drawbacks: 1) Training with (stochastic) gradient descent is slow and hard to parallelize, and SDAs take relatively long to train—even with efficient GPU implementations (Bergstra et al., 2010) and reconstruction sampling for sparse data (Dauphin et al., 2011); 2) There are several hyper-parameters (learning rate, number of epochs, noise ratio, mini-batch size and network structure), which need to be set by cross validation—this is particularly expensive as each individual run can take several hours; 3) The optimization is inherently non-convex and dependent on its initialization.

3. Marginalized Stacked Linear Denoising Autoencoders

In this section we introduce a modified version of SDA, which we refer to as *marginalized* Stacked Linear Denoising Autoencoder (mSLDA). In practice if a SDA is trained to learn features (rather than predict a target label directly), linear autoencoders are typically sufficient. Our proposed algorithm consists of stacked linear denoising autoencoders where the corruption is marginalized out in closed form — effectively yielding orders of magnitude speedups during training time. In addition, mSLDA has fewer hyper-parameters, allowing for much faster model-selection, and is layer-wise convex.

3.1 Noise Model

Similar to SDA, mSLDA learns to reconstruct the original input from its corrupted version. Therefore, we start by defining a corrupting distribution that specifies how training observations $\mathbf{x}$ are

transformed into corrupted versions $\tilde{\mathbf{x}}$. Throughout the paper, we assume a corrupting distribution of the form:

$$p(\tilde{\mathbf{x}}|\mathbf{x}) = \prod_{\alpha=1}^{d} p_E(\tilde{x}_\alpha|x_\alpha;\eta_\alpha). \tag{1}$$

where η_d is the list of user-defined hyper-parameters for the corrupting distribution. That is, we assume that 1) each dimension of the input $\mathbf{x}$ is corrupted independently; 2) the individual corrupting distributions have well-defined (finite) mean and variance, such as the Bernoulli, Poisson and Gaussian distribution. As we are going to explain later, these two assumptions leads to very efficient optimizations of our models.

For now, we are going to focus on the blank-out noise model (also often referred to as "mask-out"), which randomly sets each feature to *zero* with probability $p_\alpha \geq 0$. More precisely (with $\eta_\alpha = p_\alpha$),

$$p_E(\tilde{x}_\alpha|x_\alpha;\eta_\alpha) = \begin{cases} 0 & \text{with probability} \quad p_\alpha \\ x_\alpha & \text{with probability} \quad 1-p_\alpha \end{cases}. \tag{2}$$

Although our model is more general, for simplification we will assume that the corruption probability is identical for all features, *i.e.* $p_\alpha = p$ for all dimensions α. In Section 4, we will extend this model to different corrupting distributions.

3.2 Single-layer Denoiser

The basic building block of mSLDA is a one-layer linear denoising autoencoder. We take the unlabeled inputs $\mathbf{x}_1, \ldots, \mathbf{x}_n$ from $L \cup U$ and corrupt them with the blank-out noise, which sets each feature to 0 with probability $p \geq 0$. Let us denote the corrupted version of $\mathbf{x}_i$ as $\tilde{\mathbf{x}}_i$. As opposed to the two-level *encoder* and *decoder* in SDA, we reconstruct the corrupted inputs with a single linear mapping $\mathbf{W} : \mathcal{R}^d \rightarrow \mathcal{R}^d$, that minimizes the squared reconstruction loss

$$\frac{1}{2n}\sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{W}\tilde{\mathbf{x}}_i\|^2. \tag{3}$$

To simplify notation, we assume that a constant feature is added to the input, $\mathbf{x}_i = [\mathbf{x}_i; 1]$, and an appropriate bias is incorporated within the mapping $\mathbf{W} = [\mathbf{W}, \mathbf{b}]$. The constant feature is *never* corrupted.

The solution to (3) depends on which features of each input are randomly corrupted. To lower the variance, we perform t passes of corruption and reconstruction over the training set, each time with new randomly chosen corruptions for each input. We solve for the matrix $\mathbf{W}$ that minimizes the overall squared loss

$$\mathcal{L}_{sq}^t(\mathbf{W}) = \frac{1}{2nt}\sum_{i=1}^{n}\sum_{j=1}^{t} \|\mathbf{x}_i - \mathbf{W}\tilde{\mathbf{x}}_{i,j}\|^2, \tag{4}$$

where $\tilde{\mathbf{x}}_{i,j}$ represents the j^{th} corrupted version of the original input $\mathbf{x}_i$.

Let us define the design matrix $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathcal{R}^{d \times n}$ and its t-times repeated version as $\overline{\mathbf{X}} = [\mathbf{X}, \ldots, \mathbf{X}]$. Further, we denote the corrupted version of $\overline{\mathbf{X}}$ as $\widetilde{\mathbf{X}}$. With this notation, the loss in eq. (4) can be expressed in matrix form as

$$\mathcal{L}_{sq}^t(\mathbf{W}) = \frac{1}{2nt}\text{tr}\left[\left(\overline{\mathbf{X}} - \mathbf{W}\widetilde{\mathbf{X}}\right)^\top \left(\overline{\mathbf{X}} - \mathbf{W}\widetilde{\mathbf{X}}\right)\right]. \tag{5}$$

Algorithm 1 mLDA (for blankout corruption) in MATLAB$^{\text{TM}}$.

```
function [W,h]=mLDA(X,p);
 X=[X;ones(1,size(X,2))];
 d=size(X,1);
 q=[ones(d-1,1).*(1-p); 1];
 S=X*X';
 Q=S.*(q*q');
 Q(1:d+1:end)=q.*diag(S);
 P=S.*repmat(q',d,1);
 W=P(1:end-1,:)/(Q+1e-5*eye(d));
 h=tanh(W*X);
```

Similar to ordinary least squares (Bishop, 2006), it is straight-forward to derive a closed-form solution to (5):

$$\mathbf{W} = \mathbf{P}\mathbf{Q}^{-1} \text{ with } \mathbf{Q} = \widetilde{\mathbf{X}}\widetilde{\mathbf{X}}^\top \text{ and } \mathbf{P} = \overline{\mathbf{X}}\widetilde{\mathbf{X}}^\top. \tag{6}$$

In practice (6) can be computed as a system of linear equations, without the costly matrix inversion. (The worst-case complexity is still $O(n^3)$, but the average runtime is much accelerated.)

3.3 Marginalized Linear Denoising Autoencoder

The larger t is, the more corruptions we average over. Ideally we would like $t \to \infty$, effectively using infinitely many copies of noisy data to compute the denoising transformation $\mathbf{W}$. In this scenario, as $t \to \infty$, the loss $\mathcal{L}_{sq}$ in (4) becomes the expected reconstruction loss under $p(\tilde{\mathbf{x}}_i|\mathbf{x})$

$$\mathcal{L}_{sq}^{\infty}(\mathbf{W}) = \frac{1}{2n}\sum_{i=1}^{n} \mathbb{E}_{p(\tilde{\mathbf{x}}_i|\mathbf{x})}\left[\|\mathbf{x}_i - \mathbf{W}\tilde{\mathbf{x}}_i\|^2\right]. \tag{7}$$

We can expand this equation to obtain

$$\mathcal{L}_{sq}^{\infty}(\mathbf{W}) = \frac{1}{2n}\sum_{i=1}^{n}\left(\mathbf{x}_i\mathbf{x}_i^\top - 2\mathbf{x}_i\mathbb{E}[\tilde{\mathbf{x}}_i]^\top\mathbf{W}^\top + \mathbf{W}\mathbb{E}[\tilde{\mathbf{x}}_i\tilde{\mathbf{x}}_i^\top]\mathbf{W}^\top\right), \tag{8}$$

and, by solving for $\mathbf{W}$, the solution to (7) can then be expressed as

$$\mathbf{W} = \mathbb{E}[\mathbf{P}]\mathbb{E}[\mathbf{Q}]^{-1} \text{ with } \mathbb{E}[\mathbf{Q}] = \sum_{i=1}^{n}\mathbb{E}\left[\tilde{\mathbf{x}}_i\tilde{\mathbf{x}}_i^\top\right] \text{ and } \mathbb{E}[\mathbf{P}] = \sum_{i=1}^{n}\mathbf{x}_i\mathbb{E}[\tilde{\mathbf{x}}_i]^\top. \tag{9}$$

We refer to this algorithm as marginalized Linear Denoising Autoencoder (mLDA).

3.3.1 BLANKOUT CORRUPTION

As an example, let us consider the blankout corruption,

$$p_E(\tilde{x}_\alpha|x_\alpha;\eta_\alpha) = \begin{cases} 0 & \text{with probability} \quad p_\alpha \\ x_\alpha & \text{with probability} \quad 1-p_\alpha \end{cases}. \tag{10}$$

For notational convenience, we define a vector $\mathbf{q} = [1-p, \ldots, 1-p, 1]^\top \in \mathcal{R}^{d+1}$, where $\mathbf{q}_\alpha$ represents the probability of a feature α "surviving" the corruption. (As the constant feature is never corrupted, we have $\mathbf{q}_{d+1} = 1$.) According to the blank-out noise model defined in (10), the expected value of the corruption $\mathbb{E}[\tilde{\mathbf{x}}_i]$ can be computed as $\mathbf{x}_i \cdot \mathbf{q}$.[2] We further define the scatter matrix of the original uncorrupted input as $\mathbf{S} = \sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^\top$, and express the expectation $\mathbb{E}[\mathbf{P}]$ as

$$\mathbb{E}[\mathbf{P}] = \sum_{i=1}^{n} \mathbf{x}_i (\mathbf{x}_i \cdot \mathbf{q})^\top \text{ with } \mathbb{E}[\mathbf{P}]_{\alpha\beta} = \mathbf{S}_{\alpha\beta}\mathbf{q}_\beta. \tag{11}$$

Similarly, we can compute the expectation

$$\mathbb{E}[\mathbf{Q}] = \sum_{i=1}^{n} \mathbb{E}\left[\tilde{\mathbf{x}}_i \tilde{\mathbf{x}}_i^\top\right].$$

An off-diagonal entry in the matrix $\tilde{\mathbf{x}}_i \tilde{\mathbf{x}}_i^\top$ with index (α, β) is uncorrupted if the two features α and β both "survived" the corruption. This happens with probability $(1-p)^2$. For the diagonal entries, this holds with probability $1-p$ (because it only requires the one corresponding feature to "survived" the corruption). Thus, we can express the expectation of the matrix $\mathbf{Q}$ as

$$\mathbb{E}[\mathbf{Q}]_{\alpha,\beta} = \begin{cases} \mathbf{S}_{\alpha\beta}\mathbf{q}_\alpha\mathbf{q}_\beta & \text{if} \quad \alpha \neq \beta \\ \mathbf{S}_{\alpha\beta}\mathbf{q}_\alpha & \text{if} \quad \alpha = \beta \end{cases}. \tag{12}$$

With the help of these matrix expectations, we can compute the reconstructive mapping $\mathbf{W}$ directly in closed-form without ever explicitly constructing a single corrupted input $\tilde{\mathbf{x}}_i$. Algorithm 1 shows a 10-line MATLAB$^{\text{TM}}$ implementation of mLDA with blankout corruption. The mLDA has several advantages over traditional denoisers: 1) It requires only a single sweep through the data to compute the matrices $E[\mathbf{Q}], E[\mathbf{P}]$; 2) Training is convex and a globally optimal solution is guaranteed; 3) The optimization is performed in non-iterative closed-form.

3.4 Nonlinear feature generation and stacking

Arguably two of the key contributors to the success of the SDA are its *nonlinearity* and the *stacking* of multiple layers of denoising autoencoders to create a "deep" learning architecture. Our framework has the same capabilities.

In SDAs, the nonlinearity is injected through the nonlinear *encoder* function $h(\cdot)$, which is learned together with the reconstruction weights $\mathbf{W}$. Such an approach makes the training procedure highly non-convex and requires iterative procedures to learn the model parameters. To preserve the closed-form solution from the linear mapping in equation (5) we insert nonlinearity into our learned representation *after* the weights $\mathbf{W}$ are computed. A nonlinear squashing-function is applied on the output of each mLDA. Several choices are possible, including sigmoid, hyperbolic tangent, or the rectifier function (Nair and Hinton, 2010). Throughout this work, we use the hyperbolic tangent $\tanh()$ function and provide a detailed comparison of various squashing function in Figure 4 in Section 7.1.2.

Inspired by the layer-wise stacking of SDA, we stack several mLDA layers by feeding the output of the $(t\text{–}1)^{th}$ mLDA (after the squashing function) as the input into the t^{th} mLDA. Let us denote the

2. Here, $\mathbf{y} = \mathbf{x} \cdot \mathbf{z}$ denotes element-wise vector multiplication, *i.e.* $y_i = x_i z_i$.

Algorithm 2 mSLDA in MATLAB$^{\mathrm{TM}}$.

```
function [Ws,hs]=mSLDA(X,p,L);
 [d,n]=size(X);
 Ws=zeros(d,d+1,L);
 hs=zeros(d,n,L+1);
 hs(:,:,1)=X;
 for t=1:L
  [Ws(:,:,t), hs(:,:,t+1)]=mLDA(hs(:,:,t),p);
 end;
```

output of the t^{th} mLDA as $\mathbf{h}^t$ and the original input as $\mathbf{h}^0 = \mathbf{x}$. The training is performed greedily layer by layer: each map $\mathbf{W}^t$ is learned (in closed-form) to reconstruct the previous mLDA output $\mathbf{h}^{t-1}$ from all possible corruptions and the output of the t^{th} layer becomes $\mathbf{h}^t = \tanh(\mathbf{W}^t\mathbf{h}^{t-1})$. In our experiments, as detailed in in Section 7.1.2, we found that even without the nonlinear squashing function, stacking still improves the performance. However, the nonlinearity improves over the linear stacking significantly. We refer to the stacked denoising algorithm as marginalized Stacked Linear Denoising Autoencoders (mSLDA). Algorithm 2 shows a 8-lines MATLAB$^{\mathrm{TM}}$ implementation of mSLDA.

3.5 mSLDA for Domain Adaptation

We apply mSLDA to domain adaptation by first learning features in an unsupervised fashion on the union of the source and target data sets. One observation reported in (Glorot et al., 2011) is that if multiple domains are available, sharing the unsupervised pre-training of SDA across all domains is beneficial compared to pre-training on the source and target only. We observe a similar trend with our approach. The results reported in Section 7 are based on features learned on data from all available domains. Once a mSLDA is trained, the output of all layers, after squashing ($\tanh(\mathbf{W}^t\mathbf{h}^{t-1})$) combined with the original features $\mathbf{h}^0$, are concatenated and form the new representation. All inputs are transformed into the new feature space. A linear Support Vector Machine (SVM) (Chang and Lin, 2011) is then trained on the transformed source inputs and tested on the target domain. There are two sets of meta-parameters in mSLDA: the corruption parameters (*e.g.* p in the case of blankout corruption) and the number of layers L. In our experiments, both are set with 5-fold cross validation on the labeled data from the *source* domain. As the mSLDA training is almost instantaneous, this grid search is almost entirely dominated by the SVM training time.

4. Corruption beyond blank-out

In the previous section, we introduced mSLDA under the blank-out corruption model and derived the layer-wise closed form solution $\mathbf{W} = \mathbb{E}[\mathbf{P}]\mathbb{E}[\mathbf{Q}]^{-1}$. The derivation up to eq. (9) makes no explicit assumption on the corruption distribution and holds for any member of the exponential family with finite mean $\mathbb{E}[\tilde{\mathbf{x}}_i]$, and variance $\mathbb{V}[\tilde{\mathbf{x}}_i]$. This can be made explicit by expanding the terms $\mathbb{E}[\mathbf{P}], \mathbb{E}[\mathbf{Q}]$ as

$$\mathbb{E}[\mathbf{P}] = \sum_{i=1}^{n} \mathbf{x}_i \mathbb{E}[\tilde{\mathbf{x}}_i]^\top \text{ and } \mathbb{E}[\mathbf{Q}] = \sum_{i=1}^{n} \mathbb{E}\left[\tilde{\mathbf{x}}_i \tilde{\mathbf{x}}_i^\top\right] = \sum_{i=1}^{n} \left(\mathbb{E}[\tilde{\mathbf{x}}_i]\mathbb{E}[\tilde{\mathbf{x}}_i]^\top + \mathbb{V}[\tilde{\mathbf{x}}_i]\right). \quad (13)$$

4.1 Poisson corruption

For discrete feature values (*e.g.* word counts in a document), one interesting example of a corruption distribution is the Poisson distribution. Here, the corruption is defined as,

$$p_E(\tilde{x}_\alpha | x_\alpha; \eta_\alpha) = \frac{x_\alpha^{\tilde{x}_\alpha} e^{-x_\alpha}}{\tilde{x}_\alpha!}, \alpha = 1, \cdots, d \tag{14}$$

where the arrival rate η_α is set to x_α. In this case, we have $\mathbb{E}[\mathbf{x}] = \mathbf{x}$, and $\mathbb{V}[\mathbf{x}] = \Delta(\mathbf{x})$.[3] Note that the off-diagonal entries of the variance matrix is zero since we assume that each dimension of the input is corrupted independently. Plugging the definition (14) into eq. (13) results in

$$\mathbb{E}_{Poi}[\mathbf{P}] = \sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^\top = \mathbf{S} \text{ and } \mathbb{E}_{Poi}[\mathbf{Q}] = \sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^\top + \sum_{i=1}^{n} \Delta(\mathbf{x}_i) = \mathbf{S} + \Delta(\sum_{i=1}^{n} \mathbf{x}_i).$$

Comparing with the blank-out noise, where the corruption simulates the existence or completely absence of words due to authors' word preference, the Poisson corruption imitates different appearing frequencies for each word. Since we set the arrival rate of the distribution to be x_α, words with higher frequency in the original input will have less chance to be complete removed. In other words, we would expect the Poisson corruption to bring in less drastic change to the corrupted input $\tilde{\mathbf{x}}$ than the blank-out noise. As we can see in the experiments, the representations learned with Poisson corruption is not as robust as those with blank-out noise for domain adaptation where we would expect some words in the source domain to be completely removed from the target domain and vice versa.

4.2 Feature dependent blank-out

In Section 3.2, we introduced mLDA under the blank-out corruption model with uniform corruption rate for individual dimensions of the input. The definition of the corruption models in (1), however, allows different features to have arbitrarily different corruption rate. This enables us to incorporate prior knowledge of the corrupting distribution into our model flexibly and randomly blank-out features of different dimensions at different rate. The derivation of the two expectations $\mathbb{E}[\mathbf{P}]$ and $\mathbb{E}[\mathbf{Q}]$ is the same as in equation (11) and (12), except that a different corrupting vector $\mathbf{q}$ will be used, where each entry $\mathbf{q}_\alpha$ can take a different value.

5. Extension to High Dimensional Data

Many data sets (*e.g.* bag-of-words text documents) are naturally high dimensional and sparse. As the dimensionality increases, hill-climbing approaches used in SDAs can become prohibitively expensive. In practice, a work-around is to truncate the input data to the $r \ll d$ most common features (Glorot et al., 2011). Unfortunately, this prevents SDAs from utilizing important information found in rarer features. (As we show in Section 7, including these rarer features leads to significantly better results.) High dimensionality also poses a challenge to mSLDA, as the system of linear equations in (9) of complexity $O(d^3)$ becomes too costly. In this section we describe how to approximate this calculation with a simple division into $\frac{d}{r}$ sub-problems of $O(r^3)$.

3. Here, $\Delta(\mathbf{x})$ denotes a diagonal square matrix with $\mathbf{x}$ along its diagonal.

We combine the concept of "pivot features" from Blitzer et al. (2006) and the use of most-frequent features from Glorot et al. (2011). Instead of learning a single mapping $\mathbf{W} \in \mathcal{R}^{d\times(d+1)}$ to reconstruct all corrupted features, we learn *multiple mappings* but only reconstruct the $r \ll d$ most frequent features (here, $r = 5000$). For an input $\mathbf{x}_i$ we denote the shortened r-dimensional vector consisting of the r most-frequent features as $\mathbf{z}_i \in \mathcal{R}^r$. We divide the input features randomly into S mutually exclusive sub-sets of (roughly) equal size and learn a mapping from each one of these subsets to $\mathbf{z}_i$. Intuitively, this corresponds to "translating" rare features into common features (this is particularly successful with text documents, where the meaning of infrequent terms can often be approximated by a more frequent term.) Without loss of generality, we assume that the feature-dimensions in the input space are in random order and divide up the input vectors as $\mathbf{x}_i = \left[\mathbf{x}_i^{1\top}, \ldots, \mathbf{x}_i^{S\top}\right]^\top$. For each one of these sub-spaces we learn an independent mapping $\mathbf{W}^s$ which minimizes

$$\mathcal{L}_s(\mathbf{W}^s) = \frac{1}{2n}\sum_{i=1}^{n}\sum_{s=1}^{S}\|\mathbf{z}_i - \mathbf{W}^s\tilde{\mathbf{x}}_i^s\|^2. \tag{15}$$

Each mapping $\mathbf{W}^s$ can be solved in closed-form as in eq. (9), following the method described in section 1. We define the output of the first layer in the resulting mSLDA as the average of all reconstructions,

$$\mathbf{h}^1 = \tanh\left(\frac{1}{S}\sum_{s=1}^{S}\mathbf{W}^s\mathbf{x}^s\right). \tag{16}$$

Once the first layer of dimension $r \ll d$ is learned no further dimensionality reduction is required and we can stack subsequent layers using the regular mSLDA as described in Section 3.4 and Algorithm 2. It is worth pointing out that, although features might be separated in different sub-sets within the first layer, they can still be combined in subsequent layers of the mSLDA.

6. Alternative Formulation

In this section we want to provide the reader briefly with an alternative interpretation of mLDA with **unbiased** blank-out noise. Slightly different from the blank-out noise we introduced in Section 3.1, the unbiased version rescales the uncorrupted features to $\frac{1}{1-p}$ of their original values. More precisely (with $\eta_d = p$),

$$p_E(\tilde{x}_\alpha|x_\alpha;\eta_\alpha) = \begin{cases} 0 & \text{with probability} \quad p \\ \frac{1}{1-p}x_\alpha & \text{with probability} \quad 1-p \end{cases}. \tag{17}$$

Under this specific corruption model, and in the case where all the features are normalized to have norm 1 *across inputs*, *i.e.*, $\forall\alpha \in \{1, \cdots, d\}, \sum_{i=1}^{n} x_{i\alpha}^2 = 1$, we can then re-interpret mLDA reconstruction in eq. (7) as auto-Ridge Regression,

$$\min_{\mathbf{W}} \frac{1}{2n}\sum_{i=1}^{n}\|\mathbf{x}_i - \mathbf{W}\mathbf{x}_i\|^2 + \lambda\|\mathbf{W}\|_2^2. \tag{18}$$

with $\lambda = \frac{p}{2n(1-p)}$. In the extreme case of $p = 0$ and consequently $\lambda = 0$, the solution to (18) is trivially $\mathbf{W} = \mathbf{I}$. However, as λ increases, the $l2$ regularization encourages weights within $\mathbf{W}$ to be of comparable magnitude and reduces large diagonal entries. As p approaches 1 the regularization trade-off λ becomes ill-defined—corresponding to the pathological case where all the features are

removed in all examples, making it impossible to learn. This alternative interpretation illustrates the effect of reconstruction from blank-out corruption. Features are reconstructed from themselves and other co-occurring features and the hyper-parameter p regulates this trade-off. The effect of applying the learned reconstruction matrix to the original input, i.e., $\mathbf{Wx}$, is a smoothing of related feature values to increase robustness of the representation. In the case of domain adaptation, this facilitates some immunity over distribution drift between training and testing.

7. Experimental Results

In this section, we evaluate mSLDA on two real-world domain adaption tasks, as well as a semi-supervised learning task, and compare it with competing algorithms.

7.1 Domain adaption on text data

First, we consider a domain adaptation task for sentiment analysis. We evaluate mSLDA on the *Amazon reviews* benchmark data sets (Blitzer et al., 2006) together with several other algorithms for representation learning and domain adaptation.

Dataset. The dataset contains more than $340,000$ reviews from 25 different types of products from Amazon.com. For simplicity (and comparability), we follow the convention of (Chen et al., 2011a; Glorot et al., 2011) and only consider the binary classification problem whether a review is positive (higher than 3 stars) or negative (3 stars or lower). As mSLDA and SDA focus on feature learning, we use the raw bag-of-words (bow) unigram/bigram features as their input. To be fair to other algorithms that we compare to, we also pre-process with tf-idf (Salton and Buckley, 1988) and use the transformed feature vectors as their input if that leads to better results. Finally, we remove five domains which contain less than $1,000$ reviews.

Different domains in the complete set vary substantially in terms of number of instances and class distribution. Some domains (books and music) have hundreds of thousands of reviews, while others (food and outdoor) have only a few hundred. The proportion of negative examples in different domains also differs greatly. There are a total of 380 possible transfer tasks (*e.g.* $Apparel \rightarrow Baby$). To counter the effect of class- and size-imbalance, a more controlled smaller dataset was created by Blitzer et al. (2007), which contains reviews of four types of products: books, DVDs, electronics, and kitchen appliances. Here, each domain consists of $2,000$ labeled inputs and approximately $4,000$ unlabeled ones (varying slightly between domains) and the two classes are exactly balanced. Table 1 contains the statistics on the complete set as well as the control set. Almost all prior work provides results only on this smaller set with its more manageable *twelve* transfer tasks. We focus most of our comparative analysis on this smaller set but also provide results on the entire data for completeness.

Methods. As *baseline*, we train a linear SVM on the raw bag-of-words representation of the labeled *source* and test it on *target*. We also include the results of the same setup with dense features obtained by projecting the entire data set (labeled and unlabeled *source+target*) onto a low-dimensional sub-space with PCA (we refer to this setting as *PCA*). Besides these two baselines, we evaluate the efficacy of a linear SVM trained on features learned by mSLDA and two alternative feature learning algorithms, Structural Correspondence Learning (*SCL*) (Blitzer et al., 2006) and

DOMAIN	LABELED	UNLABELED (TEST)	NEG. INPUTS
COMPLETE (LARGE) SET			
APPAREL	4470	4470	14.52%
BABY	2046	2045	21.46%
BEAUTY	1314	1314	15.94%
BOOKS	27169	27168	12.09%
CAMERA	2652	2652	16.35%
DVDS	23044	23044	14.16%
ELECTRONICS	10197	10196	21.94%
FOOD	692	691	13.02%
GROCERY	1238	1238	13.57%
HEALTH	3254	3253	21.25%
JEWELRY	982	982	14.82%
KITCHEN	9233	9233	20.96%
MAGAZINES	1195	1195	22.64%
MUSIC	62181	62181	8.33%
OUTDOOR	729	729	20.71%
SOFTWARE	1033	1032	37.72%
SPORTS	2679	2679	18.78%
TOYS	6318	6318	19.67%
VIDEO	8695	8694	13.64%
VIDEOGAME	720	720	17.15%
CONTROLLED (SMALL) SET			
BOOKS	2000	4465	50%
DVDS	2000	3586	50%
ELECTRONICS	2000	5681	50%
KITCHEN	2000	5945	50%

Table 1: Statistics of the large and small set of the Amazon review dataset (Blitzer et al., 2007).

.

1-layer[4] *SDA* (Glorot et al., 2011). We also compare against *CODA* (Chen et al., 2011a), a state-of-the-art domain adaptation algorithm which is based on sample- and feature-selection, applied to tf-idf features. Finally, we also include a comparison with learning with Marginalized Corrupted Features (MCF) (van der Maaten et al., 2013), which also uses data corruption as a tool to improve generalization. For CODA, SDA, SCL and MCF, we use implementations provided by the authors. All hyper-parameters are set by 5-fold cross validation on the source training set[5].

Metrics. Following Glorot et al. (2011), we evaluate our results with the *transfer error* $e(S,T)$ and the *in-domain error* $e(T,T)$. The *transfer error* $e(S,T)$ denotes the classification error of a classifier trained on the labeled *source* data and tested on the unlabeled *target* data. The *in-domain*

4. We were only able to obtain the 1-layer implementation from the authors. Anecdotally, multiple-layer *SDA* implementations only lead to small improvements on this benchmark set but increase the training time drastically. The code we obtained from the authors implements the reconstruction sampling technique that was used to speed up the training of SDA for sparse inputs. While the original raw bow inputs are sparse, the output of one-layer SDA is no longer sparse, therefore, it becomes much more expensive to train.
5. We keep the default values of some of the parameters in SCL, *e.g.* the number of stop-words removed and stemming parameters — as they were already tuned for this benchmark set by the authors.

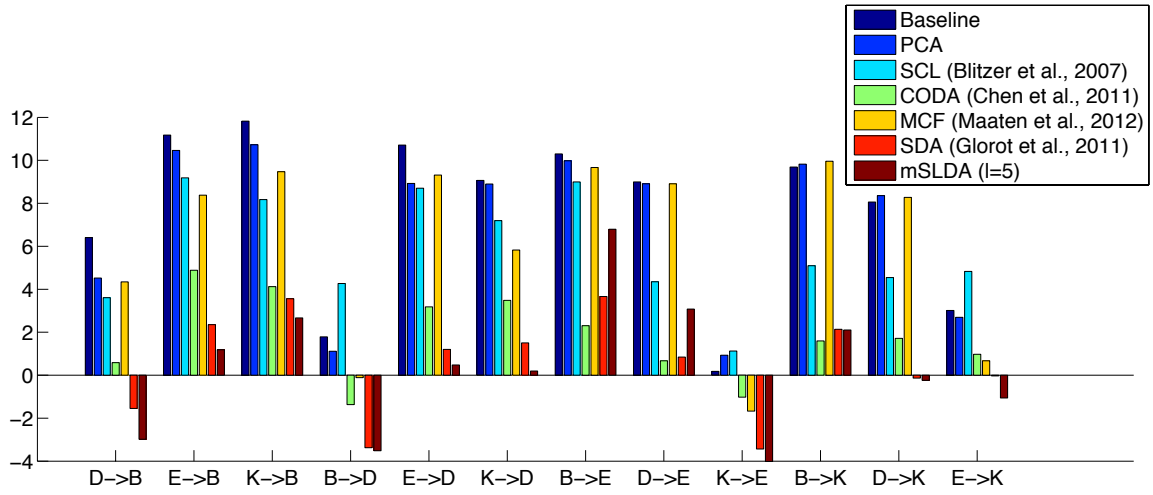


Figure 1: Comparison of mSLDA and existing works across all twelve domain adaptation task in the small Amazon review dataset.

error $e(T,T)$ denotes the classification error of a classifier that is trained on the labeled *target* data and tested on the unlabeled *target* data. Similar to Glorot et al. (2011) we measure the performance of a domain adaptation algorithm in terms of the *transfer loss*, defined as $e(S,T)-e_b(T,T)$, where $e_b(T,T)$ defines the in-domain error of the baseline (trained on the raw bow inputs). In other words, the transfer loss measures how much higher the error of an *adapted* classifier is in comparison to a linear SVM that is trained on actual *labeled target* bow data.

The various domain-adaptation tasks vary substantially in difficulty, which is why we do not average the transfer losses (which would be dominated by a few most difficult tasks). Instead, we average the *transfer ratio*, $e(S,T)/e_b(T,T)$, the ratio of the *transfer error* over the *in-domain error*. As with the *transfer loss*, a lower *transfer ratio* implies better domain adaptation.

Timing. For timing purposes, we ignore the time of the SVM training and only report the mSLDA or SDA training time.[6] As both algorithms are unsupervised, we do not re-train for different transfer tasks within a benchmark set — instead we learn one representation on the union of all domains. CODA (Chen et al., 2011b) on the other hand does not take advantage of data besides source and target. We report the average training time per transfer task.[7] All experiments were conducted on an off-the-shelf desktop with dual 6-core Intel i7 CPUs clocked at 2.66Ghz.

7.1.1 COMPARISON WITH RELATED WORK

In the first set of experiments, we use the setting from (Glorot et al., 2011) on the small Amazon benchmark set. The input data is reduced to only the $5,000$ most frequent terms of unigrams and bigrams as features.

6. As the SVM classifier is linear, we can use the extremely efficient LIBLINEAR (Fan et al., 2008) classifier, and the training time is usually in the order of seconds.
7. In CODA, the feature splitting and classifier training are inseparable and we necessarily include both in our timing.

Comparison per task. Figure 1 presents a detailed comparison of the transfer loss across the twelve domain adaptation tasks using the various methods mentioned. The reviews are from the domains *Books, Kitchen appliances, Electronics, DVDs*. Linear SVMs trained on the features generated by SDA and mSLDA clearly outperform all the other methods. Although MCF has been shown to be an effective approach for countering overfitting using noise corruption, it does not perform as well under the domain adaptation setting. As it only makes use of the training data from the source domain, it can not generalize to unseen words (terms) from the target domain. mSLDA and SDA have the advantage over CODA and MCF algorithm that they can make use of the unlabeled data from multiple source domains. For several tasks, the transfer loss becomes negative — in other words, a SVM trained on the transformed *source* data has higher accuracy than one trained on the original *target* data. (This is possible because there is more source data available. In particular, mSLDA or SDA make use of the abundant unlabeled data from multiple source domains to learn a more robust representation.) This is a strong indication that the learned new representation bridges the gap between domains. It is worth pointing out that in ten out of the twelve tasks mSLDA quickly achieves a lower transfer-loss than one-layer SDA.

Timing. Figure 2 (left) depicts the transfer ratio as a function of training time required for different algorithms, averaged over 12 tasks. It compares the results of mSLDA with the baseline, PCA, SCL, CODA and SDA. The time is plotted in log scale. We can make three observations: 1) SDA outperforms all other related work in terms of transfer-ratio, but is also the slowest to train. Note that the code we used for training SDA already implements the reconstruction sampling technique (Dauphin et al., 2011) that is specially designed to speed up the training of SDA on sparse inputs. However, as shown in the figure, it still takes more than 5 hours of training time. 2) SCL and PCA are relatively fast, but their features cannot compete in terms of transfer performance. 3) The training time of mSLDA is two orders of magnitude faster than that of SDA ($180\times$ speedup), with comparable transfer ratio. Training one layer of mLDA on all $27,677$ documents from the small set requires less than 25 seconds. A 5-layer mSLDA requires less than 2 minutes to train, and the resulting feature transformation achieves slightly better transfer ratio than a one-layer SDA.

Large scale results. To demonstrate the capabilities of mSLDA to scale to large data sets, we also evaluate it on the complete set with $n = 340,000$ reviews from 20 domains and a total of 380 domain adaptation tasks (see right plot in Figure 2). We compare mSLDA to SDA (1-layer). The large set is more heterogeneous in terms of the number of domains, domain size and class distribution than the small set. Nonetheless, a similar trend can be observed. Both the transfer error and transfer ratio are averaged across 380 tasks. The transfer ratio reported in Figure 2 (right) corresponds to averaged transfer errors of (*baseline)* 13.93%, (*one-layer SDA*) 10.50%, (*mSLDA,* $l = 1$) 11.50%, (*mSLDA,* $l = 3$) 10.47%, (*mSLDA,* $l = 5$) 10.33%. With only one layer, mSLDA performs a little worse than *SDA* but reduces the training time from over two days to about five minutes ($700\times$ speedup). With three layers, mSLDA matches the transfer-error and transfer-ratio of one-layer SDA and still only requires 14 minutes of training time ($230\times$ speedup).

7.1.2 FURTHER ANALYSIS

In addition to comparison with prior work, we also analyze various other aspects of mSLDA.

Word reconstruction As explained in Section 6, applying the learned reconstruction matrix to the original input amounts to smooth-out related feature values, which in turn helps alleviate the shift

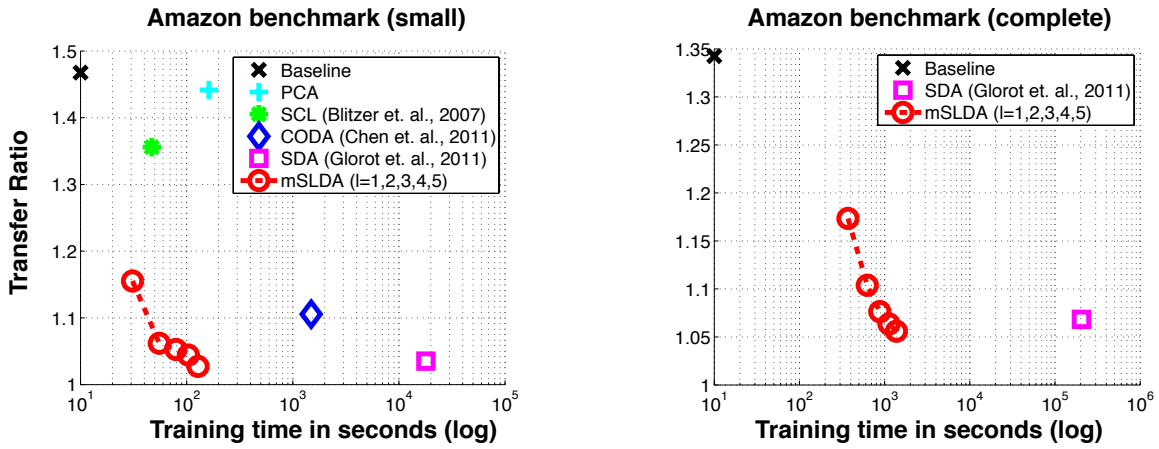


Figure 2: Transfer ratio and training times on the small (*left*) and full (*right*) Amazon Benchmark data. Results are averaged across the twelve and 380 domain adaptation tasks in the respective data sets ($5,000$ features).

between training and testing distributions. In this experiment, we apply the matrix $\mathbf{W}$ learned on the Amazon review dataset, to new input documents of a single word $\mathbf{x}$, and list the terms of the largest feature values after the smoothing $\mathbf{Wx}$. Each row of Table 2 shows an input document with a single term, and the reconstructed terms in decreasing order of feature value. As an example (*row 1*), mLDA smooths out a feature vector with a single entry at the term "great" to a denser version with values at "great for", "works great", "excellent", etc. In other words, mLDA captures word-level synonymy. The number in parentheses indicates the frequency of each word in the dataset. We can see that less frequent terms of similar meaning are reconstructed from the more frequent ones, and vice versa. As a result, a classifier trained on the smooth version of the feature vectors will be more robust, especially on rarer terms, comparing to one trained on the original sparse input.

Low-frequency features. Prior work often limits the input data to the most frequent features (Glorot et al., 2011). However there may be valuable signal in the less frequent features. We use the modification from section 5 to scale mSLDA (5-layers) up to high dimensions and include less-frequent uni-grams and bi-grams in the input (small Amazon set). In the case of SDA we make the first layer a dimensionality reducing transformation from d dimensions to 5000. The left plot in Figure 3 shows the performance of mSLDA and SDA as the input dimensionality increases (words are picked in decreasing order of their frequency). The transfer ratio is computed relative to the baseline with $d=5000$ feature. Clearly, both algorithms benefit from having more features up to $30,000$. mSLDA matches the transfer-ratio of one-layer SDA consistently and, as the dimensionality increases, gains even higher speed-up. With $30,000$ input features, SDA requires over one day and mSLDA only 3 minutes ($458\times$ speedup).

Effect of different squashing functions. Figure 4 shows the transfer ratio of mSLDA when different squashing functions are used after applying the mapping $\mathbf{W}$. We explored four different options, linear (*i.e.*, without applying any squashing function), rectifier squashing (*i.e.*, $x \rightarrow \max(0, x)$), upper bounded rectifier units (*i.e.*, $x \rightarrow \min(1, \max(0, x))$) and the hyperbolic tangent function

great(7233)	great for(484), works great(421), excellent(1697), awesome(457), easy to(1560), love it(517), great product(318), great price(183), perfect(1252), fantastic(467)
bad(2347)	horrible(511), worst(820), a bad(383), stupid(348), awful(353), terrible(593), acting(610), movie is(654), waste(1189), lame(149)
poor(1144)	poor quality(184), poorly(385), very disappointed(284), your money(637), terrible(593), very difficult(111), save your(251), disappointing(444), returned(552), waste(1189)
return(800)	returned(552), defective(263), refund(258), arrived(356), ordered(651), shipping(463), to amazon(117), i returned(221), the item(185), received(855)
fantastic(467)	love it(517), i love(1265), excellent(1697), great(7233), amazing(650), a must(364), highly recommend(560), awesome(457), i highly(379), wonderful(975), love(3066)
is amazing(141)	amazing(650), awesome(457), love it(517), a must(364), fantastic(467), great(7233), incredible(264), a wonderful(294), well worth(234), excellent(1697)
well made(136)	sturdy(314), handles(261), kitchen(784), easy to(1560), knife(314), looks great(128), pleased(503), to clean(607), stainless(320), very nice(247)
informative(133)	covers(252), an excellent(440), information(758), sections(126), helpful(368), valuable(145), guide(268), provides(372), knowledge(288), book(5523)
awkward(119)	awkward(119), to hold(309), is too(367), too small(129), disappointing(444), difficult to(489), useless(398), desk(187), impossible to(238), way too(237)

Table 2: Term reconstruction from the Amazon review dataset. Each row shows a different input term, along with terms reconstructed from this particular input in decreasing order of feature value (from left to right). The number in the parentheses indicates the frequency of each word in the dataset.

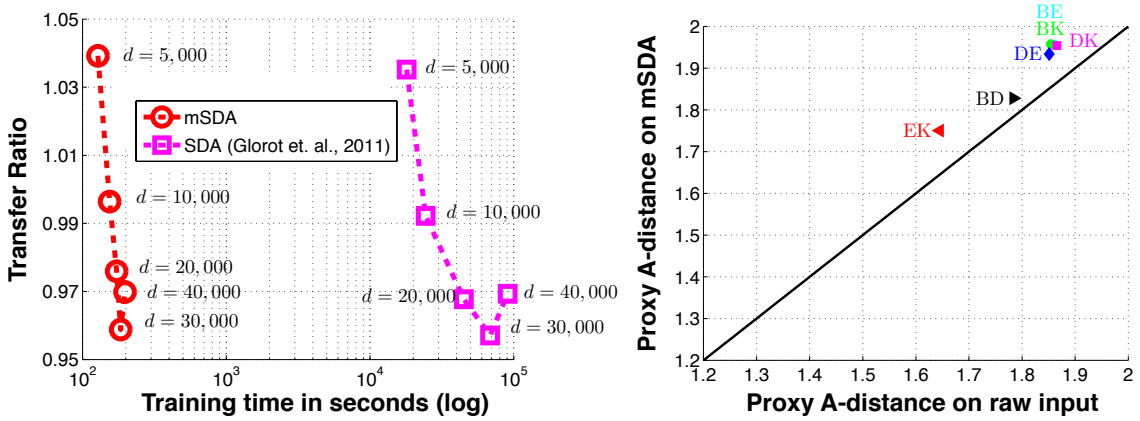


Figure 3: *Left:* Transfer ratio as a function of the input dimensionality (terms are picked in decreasing order of their frequency). *Right:* Besides domain adaptation, mSLDA *also* helps in domain *recognition* tasks.

$(x \to \tanh(x))$, which we have been using in all other experiments. The blank-out noise is used in this experiment, with the corruption level cross-validated within $[0.1, 0.9]$ of 0.1 interval. As shown in the figure, the upper bounded rectifier performs similarly as the $\tanh()$ function. For the

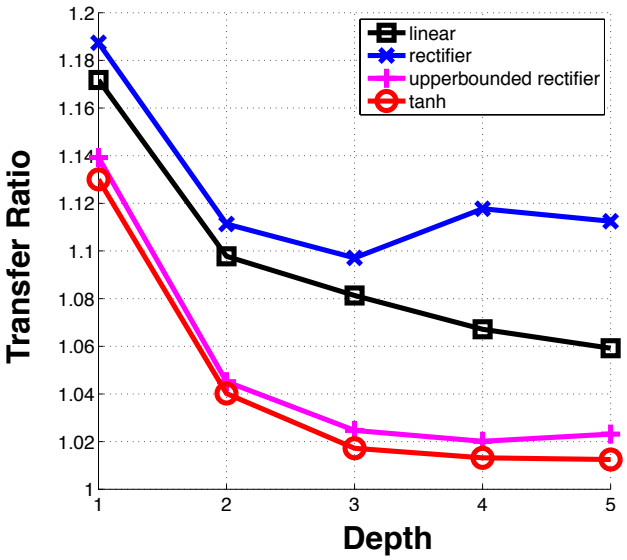


Figure 4: Transfer ratio with different squashing functions.

unbounded rectifier squashing, the performance first improves as we stack more layers, but deteriorates after three layers. The reason is that the function has no effect on large values after applying the mapping. The loss at the deeper layers is dominated by a few more frequent features while ignoring other features. One interesting observation is that even without any nonlinear squashing, the performance of mSLDA still improves as we increase the depth, as shown by the black curve in the figure.

Effect of the number of the "pivot" features. In this experiment, we investigate how the number of the "pivot" features, r, in the high-dimensional extension from Section 5 affects the performance of the algorithm. As we increase r, we would expect the transfer accuracy to be improved as well. On the other hand, since the algorithm scale cubic in term of r, the time required to solve for the mapping $\mathbf{W}$ will also increases. In the extreme case, when $r = d$, the extension reduces to our original algorithm. Figure 5 shows the transfer ratio as a function of the training time as the size of the "pivot" features increases. We do observe a reduction in transfer ratio as r increases, however, the improvement becomes marginal when r is sufficiently large (*i.e.*, 5,000). In this case, we can still finish the training of the model relatively fast (*i.e.*, within a couple of minutes).

Transfer distance. Ben-David et al. (2007) suggest the Proxy-A-distance (PAD) as a measure of how different two domains are from each other. The metric is defined as $2(1 - 2\epsilon)$, where ϵ is the generalization error of a classifier (a linear SVM in our case) trained on the binary classification problem to distinguish inputs *between* the two domains. The right plot in Figure 3 shows the PAD before and after mSLDA is applied. Surprisingly, the distance *increases* in the new representation — *i.e.* distinguishing between two domains becomes *easier* with the mSLDA features. We explain this effect through the fact that mSLDA is unsupervised and learns a generally better representation for the input data. This helps both tasks, distinguishing between domains and sentiment analysis (*e.g.* in the electronic-domain mSLDA might interpolate the feature "dvd player" from "blue ray",

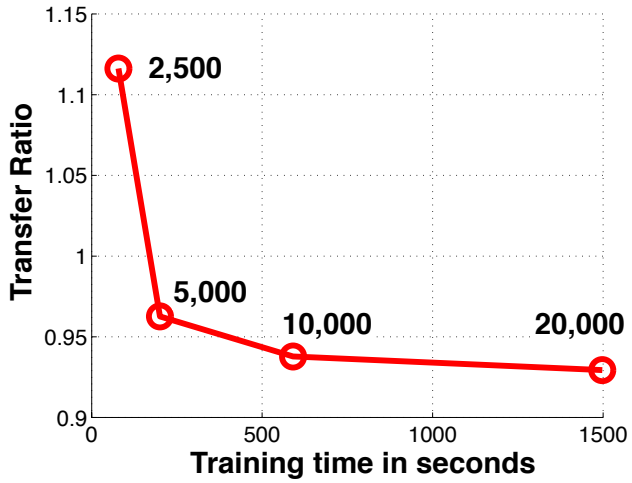


Figure 5: High dimensional extension with difference "pivot" feature size.

both are not particularly relevant for sentiment analysis but might help distinguish the review from the *book* domain.). Glorot et al. (2011) observe a similar effect with the representations learned with SDA.

Different noise model. We also apply mSLDA with the Poisson corruption model, and compare it with the blank-out noise model on the Amazon reviews data set. As shown in Figure 6, the representation learned using mSLDA with Poisson corruption also improves over the raw bag-of-word representation. Intuitively, the Poisson corruption changes the word counts and simulates the case where the same document was written with a slightly increased or decreased number occurrences of a particular word. A nice property of this corruption model is that it introduces no additional hyper-parameters. In comparison with blank-out corruption, the improvement of Poisson corruption is not as pronounced. While the Poisson corruption allows for small perturbation on the count of different words employed in the review, the blank-out noise model enables more drastic change, *i.e.*, directly removing some words. The latter scenario may reflect more closely how documents vary across domains, which results in a more robust representation. The experiment suggests that we could explore our prior knowledge on the data to properly choose the corrupting distribution used in mSLDA for better performance.

7.1.3 GENERAL TRENDS

In summary, we observe a few general trends across all experiments: 1) With one layer, mSLDA is up to three orders of magnitudes faster but slightly less expressive than the original SDA. This can be attributed to the fact that mSLDA has no hidden layer. 2) There is a clear trend that additional "stacked" layers improve the results significantly (here, up to five layers). With additional layers the mSLDA features reach (and surpass) the accuracy of 1-layer SDA and still obtain a several hundred-fold speedup. 3) The mSLDA features help diverse classification tasks, domain classification and sentiment analysis, and can be trained very efficiently on high-dimensional data.

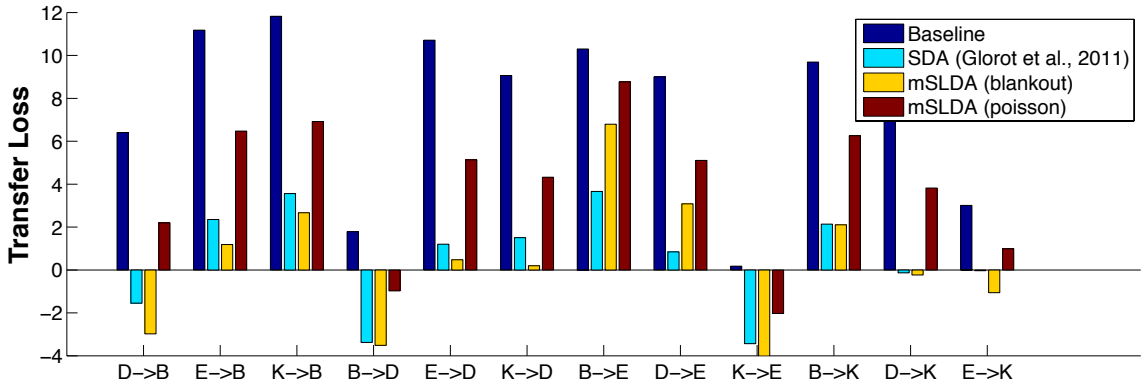


Figure 6: Comparison of mSLDA with different corruption noise in the small Amazon review dataset.

7.2 Domain adaptation on images

In this section, we evaluate mSLDA on a dataset collected by Saenko et al. (2010) for studying domain shifts in visual category recognition tasks, together with several other algorithms designed for this dataset.

Dataset. The dataset contains a total of 4,652 images of 31 categories from three domains: images from the web, images from a digital SLR camera, and images from a webcam. As shown in Figure 7, images from these domains are quite different visually. Images in the first domain are product shots downloaded from Amazon.com. The images are of medium resolution typically taken in an environment with studio lighting conditions and from a canonical viewpoint. Each category has around 90 images, capturing large intra-class variation of these categories. Images from the second domain are captured using a digital SLR camera in realistic environment with natural lighting condition. Each category has 5 different objects, and on average 3 images are captured for each object at different viewpoint. Images from the third domain are taken using a webcam. These images are of low resolution, noisy and suffer from white balance artifacts. Similar as in the second domain, 5 objects for each category are captured from different viewpoints.

Several interesting domain shifts were captured in the datasets. First, it allows us to investigate the possibility of adapting models learned on web images, which are much easier to obtain, to images captured with expensive dSLR cameras or webcams (*e.g.* mounted on robotic platforms). Second, since the same set of objects are recorded using both high-quality dSLR and the simple webcam, it allows a controlled examination of the effect of visual shift caused by different sensors.

We used the same image representation as in Saenko et al. (2010). Local scale-invariant interest points are extracted using SURF detector (Bay et al., 2006). Each image is then represented as a bag-of-visual-word with a codebook of size $d = 800$.

Our evaluation also follows the same setup as in Saenko et al. (2010). For the source domain, 8 labels per category for webcam/dSLR and 20 for amazon are available, meanwhile only three labels from the target domain are used in training as well. Five runs of experiments, each one with a set of randomly selected labels, are carried out and we report the averaged accuracies.

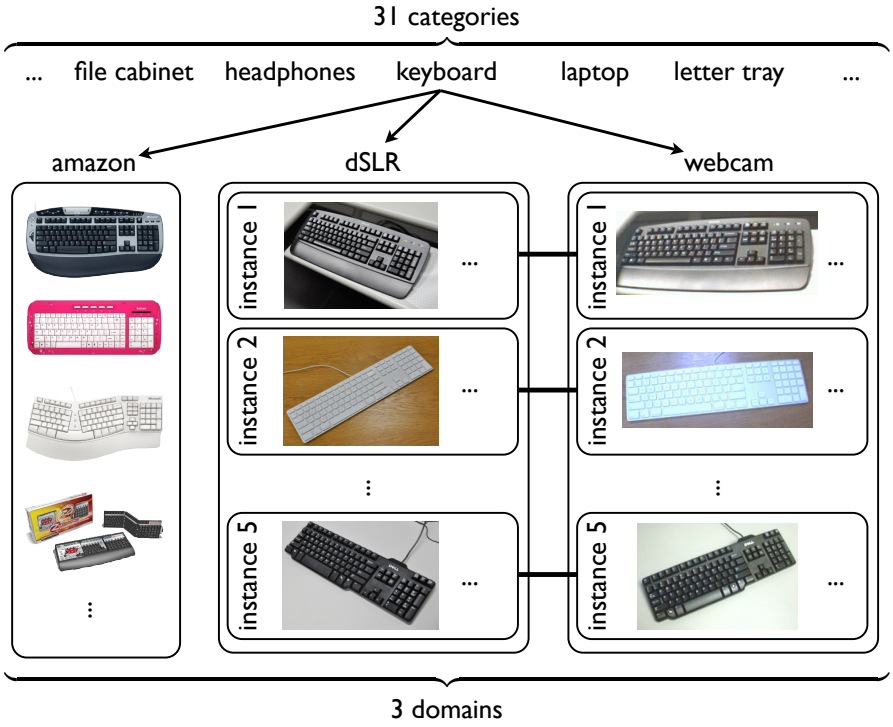


Figure 7: Sample images from the visual shift dataset. Images of objects from 31 categories are downloaded from the web as well as captured by a high definition and a low definition camera. (Saenko et al., 2010)

Methods. As *baseline*, we train a kNN model (with $k = 1$) on the raw bag-of-word representation using the *source* labeled data and test it on *target* domain (knn(A)). The same model is also trained on the combination of labeled examples from both *source* and *target* domains (knn(A+B)). We also include the results of a metric learning algorithm using information-theoretic metric learning (Davis et al., 2007). A kNN model is then trained in the projected feature space, either on all the labeled data from both domains (ITML(A+B)), or only on B labels (ITML(B)). Besides these two baselines, we also include the metric learning methods developed in Saenko et al. (2010) and its asymmetric variant by Kulis et al. (2011). For mSLDA, we present results from training both a kNN model and a linear SVM model after learning the new representation.

Table 3 summarizes the performance of these algorithms on the three domain adaptation tasks, *i.e.,* $Webcam \rightarrow dSLR$, $dSLR \rightarrow Webcam$ and $Amazon \rightarrow Webcam$. The table shows the classification test-accuracies in the target domain using various domain adaptation techniques. As we can see from comparing the two baseline algorithms, the shift between the two domains $Dslr$ and $Webcam$ is moderate since the images display the same objects and the two domains only vary in the camera resolution and lightning conditions. The adaptation between the $Amazon$ domain and $Dslr/Webcam$ involves a more drastic change, and is more challenging. mSLDA performs on par with the adapted knn methods which were especially designed on this dataset.

		BASELINE		ITML		CONSTRAINED ML		MSLDA	
SOURCE	**TARGET**	KNN(A)	KNN(A+B)	ITML(A+B)	ITML(B)	ASYMM	SYMM	KNN	LINEARSVM
WEBCAM	DSLR	.10	0.19	0.13	0.24	0.23	0.25	0.20	0.23
DSLR	WEBCAM	.26	0.28	0.20	0.27	0.28	0.29	0.31	0.38
AMAZON	WEBCAM	0.08	0.22	0.10	0.28	0.27	0.23	0.28	0.27

Table 3: Domain adaptation results (accuracy) for categories seen during training in the target domain.

7.3 Semi-supervised Learning on Text

Although mSLDA was first introduced particularly for domain adaptation, it also applies to semi-supervised learning tasks. In other words, we can use mSLDA to learn more robust representations on unlabeled data, and then train a classifier on this learned representation using labeled data only.

Dataset. We use the Reuters RCV1/RCV2 multilingual, multiview text categorization test collection (Amini et al., 2010) for evaluation. The set contains documents written in five different languages (English, French, German, Spanish and Italian) which share the same set of categories (C15, CCAT, E21, ECAT, GCAT, M11). In our experiments, we only use the subset of document that are written in English, which has 18,758 documents of vocabulary size 21,531.

Methods. As baselines, we train a linear SVM on the raw bag-of-words (*BOW*) and *TF-IDF* representations of the labeled data (Sparck Jones, 1972). In addition, we also compare against Latent semantic indexing (*LSI*) (Deerwester et al., 1990). The number of retained eigenvectors was chosen by cross-validation. For both LSI and mSLDA, we learn a new representation using the full training set (without labels), and then train a linear SVM classifier on a small subset of labeled examples using that new representation.

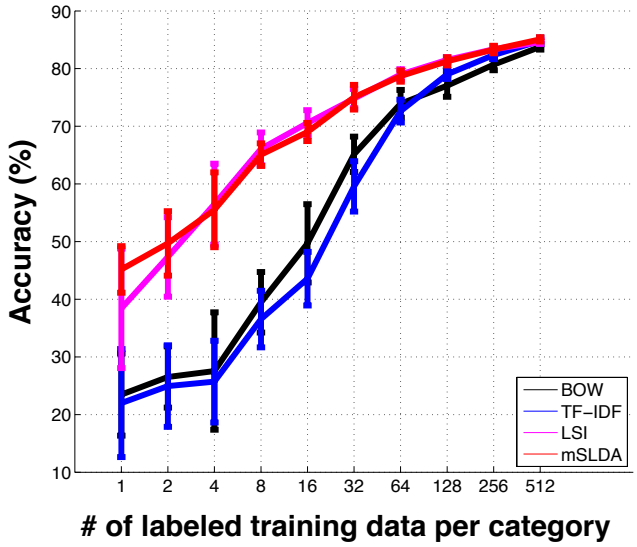


Figure 8: Semi-supervised learning results on the Reuters RCV1/RCV2 dataset.

As shown in Figure 8, we gradually increase the size of the labeled subset. For each setting, we average over 10 runs of each algorithm and report the mean accuracy as well as the variance. mSLDA performs similarly to LSI, and significantly outperform the baseline methods that were trained without unlabeled data. In summary, mSLDA learns a better representation for sparse BOW text data — however the improvement is not as pronounced as for domain adaptation. Since learning mSLDA features is cheap, it can be used as an alternative feature representation for text.

8. Discussion

In this paper we presented, mSLDA, an algorithm that marginalizes out corruption in SDA training. A key step to making this marginalization tractable, is to limit all layers within the SDA to be linear. One interesting question is to what degree this limits the expressiveness of mSLDA. As we show in our empirical results, Section 7, if mSLDA is used for feature learning, this seems to hardly matter (although more layers are necessary — something that is not really a problem as mSLDA training is so much faster.) However, the original SDA can also be used for supervised training (with fine-tuning), which is not possible with the mSLDA formulation. Maybe the fact that mSLDA works so well for bag-of-words data tells us something about the features learned by SDA. Instead of uncovering hidden concepts, as pointed out by Vincent et al. (2010), it may be more important (or simply sufficient) to learn a *common feature representation* across domains. This representation translates features from both domains into a joint space and because bag-of-words data is high dimensional, a linear mapping may just be powerful enough. Recent studies by Chen et al. (2014) seem to suggest that on more difficult image data sets the non-linear hidden representations are more important and mSLDA cannot match the performance of the original SDA.

It is an interesting observation that stacking multiple mLDA layers helps to improve these representations. One interpretation of mSLDA is to view it as a directed graph algorithm. The weight matrix $\mathbf{W}$ represents the weights of the directed edges, *i.e.* the edge from feature d to feature b has weight W_{bd}. The non-zero entries in the binary document vector $\mathbf{x}$ correspond to nodes in this graph. The transformation $\mathbf{Wx}$ takes one step in this graph, starting from the terms in $\mathbf{x}$, and accumulates the edge weights for every other term/node that is reached with this step. Stacking multiple mLDA layers is then equivalent to taking multiple consecutive steps in this fashion. Why is this helpful? Imagine two words have similar meanings but rarely co-occur. For example the terms *Obama* and *Reagan* both refer to presidents of the United States but probably rarely appear in the same sentence. If we want to perform domain adaptation from articles written in the 1980s to the 2010s, it would be good to learn that these two words refer to related entities. A single layer mLDA would learn to reconstruct co-occurring words from the term *Reagan*, such as *White House*, *President*, *United States* and it would "reconstruct" these words, but it would not reconstruct the term *Obama*. It would however also learn, from the unlabeled target data, that these very same words co-occur with the term *Obama* in more recent documents. So in the second layer it will reconstruct the word *Obama* from the terms it added in the first layer. In the graph view of mSLDA this means that *Reagan* and *Obama* are not connected through heavily weighted direct edges, but they are connected through heavily weighted two-step paths.

One interesting aspect of mSLDA is that there are only very few hyper parameters. Because training is so fast, these can be set very efficiently with cross-validation. In contrast, setting the hyper-parameters of SDA in an optimal fashion is much more time consuming. In our experiments we did not take this into account, but it is another important factor, as it may make it significantly

easier to actually find the optimal hyper-parameters for mSLDA in practice—something that will improve testing accuracy and training time alike.

Finally, although in this manuscript we primarily focused on blank-out and Poisson corruption, our proposed framework is decisively general. Different corruption distributions can be chosen for different applications, in particular when side information is available. For example, if data consists of unreliable sensor readings, then blank-out corruption could be used where the probability of blank-out is fine-tuned for each specific feature —mimicking the actual drop out rate of that particular sensor. As future work, it is also conceivable that the distribution could be learned with a generative model from the data directly.

9. Acknowledgements

We would like to thank Laurens van der Maaten for pointing out the alternative Ridge Regression formulation of mSLDA under blank-out corruption. KQW, ZX, MC were supported by NSF grants 1149882 and 1137211. FS is supported by NSF IIS-0957742, DARPA CSSG N10AP20019 and D11AP00278. The authors would also like to thank Yoshua Bengio for helpful discussions.

References

Massih R Amini, Nicolas Usunier, and Cyril Goutte. Learning from multiple partially observed views-an application to multilingual text categorization. In *Advances in neural information processing systems*, volume 1, pages 28–36, 2010.

Pierre Baldi and Kurt Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural Networks*, 2(1):53–58, 1989.

Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *Computer Vision–ECCV 2006*, pages 404–417. Springer, 2006.

Shai Ben-David, John Blitzer, Koby Crammer, Fernando Pereira, et al. Analysis of representations for domain adaptation. *Advances in Neural Information Processing Systems*, 19:137, 2007.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1-2):151–175, 2010.

James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. Theano: a cpu and gpu math expression compiler. In *Proceedings of the Python for Scientific Computing Conference (SciPy)*, volume 4, page 3. Austin, TX, 2010.

Christopher Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

John Blitzer, Ryan McDonald, and Fernando Pereira. Domain adaptation with structural correspondence learning. In *Conference on Empirical Methods in Natural Language Processing*, Sydney, Australia, 2006.

John Blitzer, Mark Dredze, and Fernando Pereira. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Association for Computational Linguistics*, 2007.

Christopher JC Burges and Bernhard Schölkopf. Improving the accuracy and speed of support vector machines. In *Advances in Neural Information Processing Systems*, pages 375–381, 1997.

Chih-Chung Chang and Chih-Jen Lin. Libsvm: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

Olivier Chapelle, Pannagadatta Shivaswamy, Srinivas Vadrevu, Kilian Weinberger, Ya Zhang, and Belle Tseng. Boosted multi-task learning. *Machine learning*, 85(1-2):149–173, 2011.

Minmin Chen, Yixin Chen, and Kilian Q Weinberger. Automatic feature decomposition for single view co-training. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 953–960, 2011a.

Minmin Chen, Kilian Q Weinberger, and John Blitzer. Co-training for domain adaptation. In *Advances in Neural Information Processing Systems*, pages 2456–2464, 2011b.

Minmin Chen, Zhixiang Xu, Fei Sha, and Kilian Q Weinberger. Marginalized denoising autoencoders for domain adaptation. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pages 767–774, 2012.

Minmin Chen, Kilian Q Weinberger, Fei Sha, and Yoshua Bengio. Marginalized denoising autoencoders for nonlinear representations. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1476–1484, 2014.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

Hal Daume III. Frustratingly easy domain adaptation. In *Association for Computational Linguistics*, page 256, 2007.

Yann N Dauphin, Xavier Glorot, and Yoshua Bengio. Large-scale learning of embeddings with reconstruction sampling. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 945–952, 2011.

Jason V Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S Dhillon. Information-theoretic metric learning. In *Proceedings of the 24th international conference on Machine learning*, pages 209–216. ACM, 2007.

Scott C Deerwester, Susan T Dumais, Thomas K Landauer, George W Furnas, and Richard A Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 513–520, 2011.

Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *Computer Vision and Pattern Recognition (CVPR)*, pages 2066–2073. IEEE, 2012.

Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

Jiayuan Huang, Arthur Gretton, Karsten M Borgwardt, Bernhard Schölkopf, and Alex J Smola. Correcting sample selection bias by unlabeled data. In *Advances in Neural Information Processing Systems*, pages 601–608, 2006.

Koray Kavukcuoglu, Marc Aurelio Ranzato, Rob Fergus, and Yann Le-Cun. Learning invariant features through topographic filter maps. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1605–1612. IEEE, 2009.

Brian Kulis, Kate Saenko, and Trevor Darrell. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1785–1792. IEEE, 2011.

Honglak Lee, Yan Largman, Peter Pham, and Andrew Y Ng. Unsupervised feature learning for audio classification using convolutional deep belief networks. *Advances in neural information processing systems*, 22:1096–1104, 2009.

Qian Liu, Aaron Mackey, David Roos, and Fernando Pereira. Evigan: a hidden variable model for integrating gene evidence for eukaryotic gene prediction. *Bioinformatics*, 2008.

Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation with multiple sources. In *Advances in Neural Information Processing Systems*, pages 1041–1048, 2009.

David McClosky, Eugene Charniak, and Mark Johnson. Reranking and self-training for parser adaptation. In *Proceedings of the 44th Association for Computational Linguistics*, pages 337–344. Association for Computational Linguistics, 2006.

Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 807–814, 2010.

Salah Rifai, Pascal Vincent, Xavier Muller, Xavier Glorot, and Yoshua Bengio. Contractive auto-encoders: Explicit invariance during feature extraction. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 833–840, 2011.

David E Rumelhart, Geoffery E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *Computer Vision–ECCV 2010*, pages 213–226. Springer, 2010.

Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988.

Karen Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21, 1972.

Laurens van der Maaten, Minmin Chen, Stephen Tyree, and Kilian Weinberger. Learning with marginalized corrupted features. In *Proceedings of the International Conference on Machine Learning*, 2013.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM, 2008.

Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 9999:3371–3408, 2010.

Sida Wang and Christopher Manning. Fast dropout training. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 118–126, 2013.

Kilian Weinberger, Anirban Dasgupta, John Langford, Alex Smola, and Josh Attenberg. Feature hashing for large scale multitask learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1113–1120. ACM, 2009.

Zhixiang Eddie Xu, Minmin Chen, Kilian Weinberger, and Fei Sha. From sbow to dcot marginalized encoders for text representation. In *CIKM*, pages 1879–1884, 2012.

Gui-Rong Xue, Wenyuan Dai, Qiang Yang, and Yong Yu. Topic-bridged plsa for cross-domain text classification. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 627–634. ACM, 2008.

Journal of Machine Learning Research 16 (2015) 3877-3903 Submitted 12/13; Revised 2/15; Published 12/15

PAC Optimal MDP Planning with Application to Invasive Species Management*

Majid Alkaee Taleghan ALKAEE@EECS.OREGONSTATE.EDU
Thomas G. Dietterich TGD@EECS.OREGONSTATE.EDU
Mark Crowley CROWLEY@EECS.OREGONSTATE.EDU
School of Electrical Engineering and Computer Science, Oregon State University, Corvallis, OR 97331

Kim Hall KIM.HALL@OREGONSTATE.EDU
Department of Forest Ecosystems and Society, Oregon State University, Corvallis, OR 97331

H. Jo Albers JO.ALBERS@UWYO.EDU
Haub School of Environment and Natural Resources and Department of Economics and Finance, University of Wyoming, Laramie, WY 82072

Editor: Peter Auer, Marcus Hutter, and Laurent Orseau

Abstract

In a simulator-defined MDP, the Markovian dynamics and rewards are provided in the form of a simulator from which samples can be drawn. This paper studies MDP planning algorithms that attempt to minimize the number of simulator calls before terminating and outputting a policy that is approximately optimal with high probability. The paper introduces two heuristics for efficient exploration and an improved confidence interval that enables earlier termination with probabilistic guarantees. We prove that the heuristics and the confidence interval are sound and produce with high probability an approximately optimal policy in polynomial time. Experiments on two benchmark problems and two instances of an invasive species management problem show that the improved confidence intervals and the new search heuristics yield reductions of between 8% and 47% in the number of simulator calls required to reach near-optimal policies.

Keywords: invasive species management, Markov decision processes, MDP planning, Good-Turing estimate, reinforcement learning

1. Introduction

The motivation for this paper is the area of ecosystem management in which a manager seeks to maintain the healthy functioning of an ecosystem by taking actions that promote the persistence and spread of endangered species or actions that fight the spread of invasive species, fires, and disease. Most ecosystem management problems can be formulated as MDP (Markov Decision Process) planning problems with separate planning and execution phases. During the planning phase, the algorithm can invoke a simulator to obtain samples of the transitions and rewards. Simulators in these problems typically model the system to high fidelity and, hence, are very expensive to execute. Consequently, the time required to solve such MDPs is dominated by the number of calls to

*. Portions of this work appeared in Proceedings of Twenty-Seventh AAAI Conference on Artificial Intelligence (AAAI-2013)

the simulator. A good MDP planning algorithm minimizes the number of calls to the simulator and yet terminates with a policy that is approximately optimal with high probability. This is referred to as being PAC-RL (Fiechter, 1994).

Because of the separation between the exploration phase (where the simulator is invoked and a policy is computed) and the exploitation phase (where the policy is executed in the actual ecosystem), we refer to these ecosystem management problems as problems of *MDP Planning* rather than of *Reinforcement Learning*. In MDP planning, we do not need to resolve the exploration-exploitation tradeoff.

Another aspect of these MDP planning problems that distinguishes them from reinforcement learning is that the planning algorithm must decide when to terminate and output a PAC-optimal policy. Many reinforcement learning algorithms, such as Sparse Sampling (Kearns et al., 1999), FSSS (Walsh et al., 2010), MBIE (Strehl and Littman, 2008), and UCRL2 (Jaksch et al., 2010) never terminate. Instead, their performance is measured in terms of the number of "significantly non-optimal actions" (known as PAC-MDP, Kakade (2003)) or cumulative regret (Jaksch et al., 2010).

A final aspect of algorithms for ecosystem management problems is that they must produce an explicit policy in order to support discussions with stakeholders and managers to convince them to adopt and execute the policy. Hence, receding horizon search methods, such as Sparse Sampling and FSSS, are not appropriate because they do not compute an explicit policy.

A naive approach to solving simulator-defined MDP planning problems is to invoke the simulator a sufficiently large number of times in every state-action pair and then apply standard MDP planning algorithms to compute a PAC-optimal policy. While this is required in the worst case (c.f., Azar et al. (2012)), there are two sources of constraint that algorithms can exploit to reduce simulator calls. First, the transition probabilities in the MDP may be sparse so that only a small fraction of states are directly reachable from any given state. Second, in MDP planning problems, there is a designated starting state s_0, and the goal is to find an optimal policy for acting in that state and in all states *reachable* from that state. In the case where the optimality criterion is cumulative *discounted* reward, an additional constraint is that the algorithm only need to consider states that are reachable within a fixed horizon, because rewards far in the future have no significant impact on the value of the starting state.

It is interesting to note that the earliest PAC-optimal algorithm published in the reinforcement learning community was in fact an MDP planning algorithm: the method of Fiechter (1994) addresses exactly the problem of making a polynomial number of calls to the simulator and then outputting a policy that is approximately correct with high probability. Fiechter's method works by exploring a series of trajectories, each of which begins at the start state and continues to a fixed-depth horizon. By exploring along trajectories, this algorithm ensures that only reachable states are explored. And by terminating the exploration at a fixed horizon, it exploits discounting.

Our understanding of reinforcement learning has advanced considerably since Fiechter's work. This paper can be viewed as applying these advances to develop "modern" MDP planning algorithms. Specifically, we introduce the following five improvements:

1. Instead of exploring along trajectories, we take advantage of the fact that our simulators can be invoked for any state-action pair in any order. Hence, our algorithms perform fine-grained exploration where they iteratively select the state-action pair that they believe will be most informative.

2. By not exploring along trajectories (rooted at the start state), we could potentially lose the guarantee that the algorithm only explores states that are reachable from the start state. We address this by maintaining an estimate of the discounted state occupancy measure. This measure is non-zero only for states reachable from the start state. We also use the occupancy measure in our exploration heuristics.

3. We adopt an extension to the termination condition introduced by Even-Dar et al. (2002, 2006), which is the width of a confidence interval over the optimal value of the start state. We halt when the width of the confidence interval is less than ε, the desired accuracy bound.

4. We replace the Hoeffding-bound confidence intervals employed by Fiechter (and others) with the multinomial confidence intervals of Weissman, Ordentlich, Seroussi, Verdu, and Weinberger (2003) employed in the MBIE algorithm of Strehl and Littman (2008).

5. To take advantage of sparse transition functions, we incorporate an additional confidence interval for the Good-Turing estimate of the "missing mass" (the total probability of all unobserved outcomes for a given state-action pair). This confidence interval can be easily combined with the Weissman et al. interval.

This paper is organized as follows. Section 2 introduces our notation. Section 3 describes a particular ecosystem management problem—control of the invasive plant tamarisk—and its formulation as an MDP. Section 4 reviews previous work on sample-efficient MDP planning and describes in detail the algorithms against which we will evaluate our new methods. Section 5 presents the technical contributions of the paper. It introduces our improved confidence intervals, proves their soundness, and presents experimental evidence that they enable earlier termination than existing methods. It then describes two new exploration heuristics, proves that they achieve polynomial sample size, and presents experimental evidence that they are more effective than previous heuristics. Section 6 concludes the paper.

2. Definitions

We employ the standard formulation of an infinite horizon discounted Markov Decision Process (MDP; Bellman 1957; Puterman 1994) with a designated start state distribution. Let the MDP be defined by $\mathscr{M} = \langle S, A, P, R, \gamma, P_0 \rangle$, where S is a finite set of (discrete) states of the world; A is a finite set of possible actions that can be taken in each state; $P : S \times A \times S \mapsto [0,1]$ is the conditional probability of entering state s' when action a is executed in state s; $R(s,a)$ is the (deterministic) reward received after performing action a in state s; $\gamma \in (0,1)$ is the discount factor, and P_0 is the distribution over starting states. It is convenient to define a special starting state s_0 and action a_0 and define $P(s|s_0,a_0) = P_0(s)$ and $R(s_0,a_0) = 0$. We assume that $0 \leq R(s,a) \leq R_{max}$ for all s,a. Generalization of our methods to (bounded) stochastic rewards is straightforward.

A *strong simulator* (also called a *generative model*) is a function $F : S \times A \mapsto S \times \Re$ that given (s,a) returns (s',r) where s' is sampled according to $P(s'|s,a)$ and $r = R(s,a)$.

A (deterministic) policy is a function from states to actions, $\pi : S \mapsto A$. The value of a policy π at the starting state is defined as $V^\pi(s_0) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t R(s_t, \pi(s_t))]$, where the expectation is taken with respect to the stochastic transitions. The maximum possible $V^\pi(s_0)$ is denoted $V_{max} = R_{max}/(1-\gamma)$. An optimal policy π^* maximizes $V^\pi(s_0)$, and the corresponding value is denoted by

$V^*(s_0)$. The action-value of state s and action a under policy π is defined as $Q^\pi(s,a) = R(s,a) + \gamma \sum_{s'} P(s'|s,a) V^\pi(s')$. The optimal action-value is denoted $Q^*(s,a)$.

Define $pred(s)$ to be the set of states s^- such that $P(s|s^-,a) > 0$ for at least one action a and $succ(s,a)$ to be the set of states s' such that $P(s'|s,a) > 0$.

Definition 1 *Fiechter (1994). A learning algorithm is PAC-RL*[1] *if for any discounted MDP defined by* $\langle S,A,P,R,\gamma,P_0 \rangle$, $\varepsilon > 0$, $1 > \delta > 0$, *and* $0 \leq \gamma < 1$, *the algorithm halts and outputs a policy* π *such that*

$$\mathbb{P}[|V^*(s_0) - V^\pi(s_0)| \leq \varepsilon] \geq 1 - \delta,$$

in time polynomial in $|S|$, $|A|$, $1/\varepsilon$, $1/\delta$, $1/(1-\gamma)$, *and* R_{max}.

As a learning algorithm explores the MDP, it collects the following statistics. Let $N(s,a)$ be the number of times the simulator has been called with state-action pair (s,a). Let $N(s,a,s')$ be the number of times that s' has been observed as the result. Let $R(s,a)$ be the observed reward.

3. Managing Tamarisk Invasions in River Networks

The tamarisk plant (*Tamarix* spp.) is a native of the Middle East. It has become an invasive plant in the dryland rivers and streams of the western US (DiTomaso and Bell, 1996; Stenquist, 1996). It out-competes native vegetation primarily by producing large numbers of seeds. Given an ongoing tamarisk invasion, a manager must repeatedly decide how and where to fight the invasion (e.g., eradicate tamarisk plants? plant native plants? upstream? downstream?).

A stylized version of the tamarisk management problem can be formulated as an MDP as follows. The state of the MDP consists of a tree-structured river network in which water flows from the leaf nodes toward the root (see Figure 1). The network contains E edges. Each edge in turn has H slots at which a plant can grow. Each slot can be in one of three states: empty, occupied by a tamarisk plant, or occupied by a native plant. In this stylized model, because the exact physical layout of the H slots within each edge is unimportant, the state of the edge can be represented using only the number of slots that are occupied by tamarisk plants and the number of slots occupied by native plants. The number of empty slots can be inferred by subtracting these counts from H. Hence, each edge can be in one of $(H+1)(H+2)/2$ states. Consequently, the total number of states in the MDP is $E^{(H+1)(H+2)/2}$.

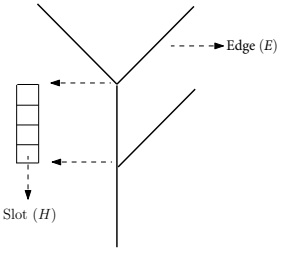


Figure 1: Tamarisk structure

The dynamics are defined as follows. In each time step, each plant (tamarisk or native) dies with probability 0.2. The remaining plants each produce 100 seeds. The seeds then disperse according to a spatial process such that downstream spread is much more likely than upstream spread. We employ the dispersal model of Muneepeerakul et al. (2007, Appendix B) with an upstream parameter of 0.1 and a downstream parameter of 0.5. An important aspect of the dispersal model is that there is a

1. In retrospect, it would have been better if Fiechter had called this PAC-MDP, because he is doing MDP planning. In turn, PAC-MDP has come to refer to *reinforcement learning* algorithms with polynomial time or regret bounds, which would be more appropriately called PAC-RL algorithms. At some point, the field should swap the meaning of these two terms.

non-zero probability for a propagule to travel from any edge to any other edge. Each propagule that arrives at an edge lands in a slot chosen uniformly at random. Hence, after dispersal, each propagule has landed in one of the slots in the river network. The seeds that arrive at an occupied slot die and have no effect. The seeds that arrive at an empty slot compete stochastically to determine which one will occupy the site and grow. In the MDPs studied in this paper, this competition is very simple: one of the arriving seeds is chosen uniformly at random to occupy the slot.

Many variations of the model are possible. For example, we can allow the tamarisk plants to be more fecund (i.e., produce more seeds) than the native plants. The seeds can have differential competitive advantage. The plants can have differential mortality, and so on. One variation that we will employ in one of our experiments is to include "exogenous arrivals" of tamarisk seeds. This models the process by which new seeds are introduced to the river network from some external source (e.g., fishermen transporting seeds on their clothes or equipment). Specifically, in the exogenous arrivals condition, in addition to the seeds that arrive at an edge via dispersal, up to 10 additional seeds of each species arrive in each edge. These are sampled by taking 10 draws from a Bernoulli distribution for each species. For tamarisk, the Bernoulli parameter is 0.1; for the native seeds, the Bernoulli parameter is 0.4.

The dynamics can be represented as a very complex dynamic Bayesian network (DBN). However, inference in this DBN is intractable, because the induced tree width is immense. One might hope that methods from the factored MDP literature could be applied, but the competition between the seeds that arrive at a given slot means that every slot is a parent of every other slot, so there is no sparseness to be exploited. An additional advantage of taking a simulation approach is that our methods can be applied to any simulator-defined MDP. We have therefore constructed a simulator that draws samples from the DBN. Code for the simulator can be obtained from `http://2013.rl-competition.org/domains/invasive-species`.

The actions for the management MDP are defined as follows. At each time step, one action can be taken in each edge. The available actions are "do nothing", "eradicate" (attempt to kill all tamarisk plants in all slots in the edge), and "restore" (attempt to kill all tamarisk plants in all slots in the edge and then plant native plants in every empty slot). The effects are controlled by two parameters: the probability that killing a tamarisk plant succeeds ($\chi = 0.85$) and the probability that planting a native plant in an empty slot succeeds ($\beta = 0.65$). Taken together, the probability that the "restore" action will change a slot from being occupied by a tamarisk plant to being occupied by a native plant is the product $\chi \times \beta = 0.5525$. Because these actions can be taken in each edge, the total number of actions for the MDP is 3^E. However, we will often include a budget constraint that makes it impossible to treat more than one edge per time step.

The reward function assigns costs as follows. There is a cost of 1.0 for each edge that is invaded (i.e., that has at least one slot occupied by a tamarisk plant) plus a cost of 0.1 for each slot occupied by a tamarisk plant. The cost of applying an action to an edge is 0.0 for "do nothing", 0.5 for "eradicate", and 0.9 for "restore".

The optimization objective is to minimize the infinite horizon discounted sum of costs. However, for notational consistency we will describe our algorithms in terms of maximizing the discounted sum of rewards throughout the paper.

It is important to note that in real applications, all of the parameters of the cost function and transition dynamics may be only approximately known, so another motivation for developing sample-efficient algorithms is to permit experimental analysis of the sensitivity of the optimal policy to the

values of these parameters. The techniques employed in this paper are closely-related to those used to compute policies that are robust to these uncertainties (Mannor et al., 2012; Tamar et al., 2014).

Now that we have described our motivating application problem, we turn our attention to developing efficient MDP planning algorithms. We start by summarizing previous research.

4. Previous Work on Sample-Efficient MDP Planning

Fiechter (1994) first introduced the notion of PAC reinforcement learning in Definition 1 and presented the PAC-RL algorithm shown in Figure 1. Fiechter's algorithm defines a measure of uncertainty $\tilde{d}_h^{\pi}(s)$, which with high probability is an upper bound on the difference $|V_h^*(s) - V_h^{\pi}(s)|$ between the value of optimal policy and the value of the "maximum likelihood" policy that would be computed by value iteration using the current transition probability estimates. The subscript h indicates the depth of state s from the starting state. Fiechter avoids dealing with loops in the MDP by computing a separate transition probability estimate for each combination of state, action and depth (s,a,h) up to $h \leq H$, where H is the maximum depth ("horizon") at which estimates are needed. Hence, the algorithm maintains separate counts $N_h(s,a,s')$ and $N_h(s,a)$ to record the results of exploration for each depth h. To apply this algorithm in practice, Fiechter (1997) modifies the algorithm to drop the dependency of the related statistics on h.

Fiechter's algorithm explores along a sequence of trajectories. Each trajectory starts at state s_0 and depth 0 and follows an exploration policy π^e until reaching depth H. The exploration policy is the optimal policy for an "exploration MDP" whose transition function is $P_h(s'|s,a)$ but whose reward function for visiting state s at depth h is equal to

$$R_h(s,a) = \frac{6}{\varepsilon}\frac{V_{max}}{1-\delta}\sqrt{\frac{2\ln 4H|S||A| - 2\ln\delta}{N_h(s,a)}}.$$

This reward is derived via an argument based on the Hoeffding bound. The transition probabilities $P_h(s'|s,a)$ are computed from the observed counts.

The quantity $d^{\pi^e}(s)$ is the value function corresponding to π^e. Because the MDP is stratified by depth, π^e and d^{π_e} can be computed in a single sweep starting at depth H and working backward to depth 0. The algorithm alternates between exploring along a single trajectory and recomputing π^e and d^{π^e}. It halts when $d_0^{\pi^e}(s_0) \leq 2/(1-\gamma)$. By exploring along π^e, the algorithm seeks to visit a sequence of states whose total uncertainty is maximized in expectation.

A second important inspiration for our work is the Model-Based Action Elimination (MBAE) algorithm of Even-Dar et al. (2002, 2006). Their algorithm maintains confidence intervals $Q(s,a) \in [Q_{lower}(s,a), Q_{upper}(s,a)]$ on the action-values for all state-action pairs in the MDP. These confidence intervals are computed via "extended value iteration" that includes an additional term derived from the Hoeffding bounds:

$$Q_{upper}(s,a) = R(s,a) + \gamma\sum_{s'}\hat{P}(s'|s,a)V_{upper}(s') + V_{max}\sqrt{\frac{\ln ct^2|S||A| - \ln\delta}{|N(s,a)|}} \quad (1)$$

$$V_{upper}(s) = \max_a Q_{upper}(s,a) \quad (2)$$

$$Q_{lower}(s,a) = R(s,a) + \gamma\sum_{s'}\hat{P}(s'|s,a)V_{lower}(s') - V_{max}\sqrt{\frac{\ln ct^2|S||A| - \ln\delta}{|N(s,a)|}} \quad (3)$$

$$V_{lower}(s) = \max_a Q_{lower}(s,a). \quad (4)$$

Algorithm 1: Fiechter($s_0, \gamma, F, \varepsilon, \delta$)

Input: s_0: start state; γ: discount rate; F: a simulator
Initialization:
  $H = \left\lceil \frac{1}{1-\gamma}\left(\ln V_{max} + \ln \frac{6}{\varepsilon}\right)\right\rceil$ // horizon depth
  for $s, s' \in S, a \in A(s), h = 0, \ldots, H-1$ **do**
    $N_h(s,a) = 0$
    $N_h(s,a,s') = 0$
    $R_h(s,a,s') = 0$
    $\pi^e_h(s) = a_1$
Exploration:
  while $d^{\pi^e}_0(s_0) > 2/(1-\gamma)$ **do**
    reset $h = 0$ and $s = s_0$
    while $h < H$ **do**
      $a = \pi^e_h(s)$
      $(r, s') \sim F(s,a)$ // draw sample
      update $N_h(s,a)$, $N_h(s,a,s')$, and $R_h(s,a,s')$
      $h = h + 1$
      $s = s'$
    Compute new policy π^e (and values d^{π^e}) using the following dynamic program
    $d_{max} = (12 V_{max})/(\varepsilon(1-\gamma))$
    $P_h(s'|s,a) = N_h(s,a,s')/N_h(s,a)$
    $d^{\pi^e}_H(s) = 0, \forall s \in S$
    for $h = H-1, \ldots, 0$ **do**
      $e^{\pi^e}_h(s,a) = \min\left\{d_{max}, \frac{6}{\varepsilon}\frac{V_{max}}{1-\delta}\sqrt{\frac{2\ln 4H|S||A| - 2\ln\delta}{N_h(s,a)}} + \gamma \sum_{s' \in succ(s,a)} P_h(s'|s,a) d^{\pi^e}_{h+1}(s')\right\}$
      $\pi^e_h(s) = \text{argmax}_{a \in A(s)}\, e^{\pi^e}_h(s,a)$
      $d^{\pi^e}_h(s) = e^{\pi^e}_h(s, \pi^e_h(s))$
Compute policy π, and return it.

In these equations, t is a counter of the number of times that the confidence intervals have been computed and c is an (unspecified) constant. Even-Dar et al. prove that the confidence intervals are sound. Specifically, they show that with probability at least $1-\delta$, $Q_{lower}(s,a) \leq Q^*(s,a) \leq Q_{upper}(s,a)$ for all s, a, and iterations t.

Their MBAE algorithm does not provide a specific exploration policy. Instead, the primary contribution of their work is to demonstrate that these confidence intervals can be applied as a termination rule. Specifically, if for all (s,a), $|Q_{upper}(s,a) - Q_{lower}(s,a)| < \frac{\varepsilon(1-\gamma)}{2}$, then the policy that chooses actions to minimize $Q_{lower}(s,a)$ is ε-optimal with probability at least $1-\delta$. Note that the iteration over s' in these equations only needs to consider the observed transitions, as $\hat{P}(s'|s,a) = 0$ for all transitions where $N(s,a,s') = 0$.

An additional benefit of the confidence intervals is that any action a' can be eliminated from consideration in state s if $Q_{upper}(s,a') < Q_{lower}(s,a)$. Even-Dar et al. demonstrate experimentally

that this can lead to faster learning than standard Q learning (with either uniform random action selection or ε-greedy exploration).

The third important source of ideas for our work is the Model-Based Interval Estimation (MBIE) algorithm of Strehl and Littman (2008). MBIE maintains an upper confidence bound on the action-value function, but unlike Fiechter and Even-Dar et al., this bound is based on a confidence region for the multinomial distribution developed by Weissman et al. (2003).

Let $\hat{P}(s'|s,a) = N(s,a,s')/N(s,a)$ be the maximum likelihood estimate for $P(s'|s,a)$, and let $\hat{P}$ and $\tilde{P}$ denote $\hat{P}(\cdot|s,a)$ and $\tilde{P}(\cdot|s,a)$. Define the confidence set CI as

$$CI(\hat{P}|N(s,a),\delta) = \left\{\tilde{P} \;\middle|\; \|\tilde{P}-\hat{P}\|_1 \leq \omega(N(s,a),\delta)\right\}, \tag{5}$$

where $\|\cdot\|_1$ is the L_1 norm and $\omega(N(s,a),\delta) = \sqrt{\frac{2[\ln(2^{|S|}-2)-\ln\delta]}{N(s,a)}}$. The confidence interval is an L_1 "ball" of radius $\omega(N(s,a),\delta)$ around the maximum likelihood estimate for P. Weissman et al. (2003) prove that with probability $1-\delta$, $P(\cdot|s,a) \in CI(\hat{P}(\cdot|s,a)|N(s,a),\delta)$.

Given confidence intervals for all visited (s,a), MBIE computes an upper confidence bound on Q and V as follows. For any state where $N(s,a) = 0$, define $Q_{upper}(s,a) = V_{max}$. Then iterate the following dynamic programming equations to convergence:

$$Q_{upper}(s,a) = R(s,a) + \max_{\tilde{P}(s,a)\in CI(P(s,a),\delta_1)} \gamma \sum_{s'} \tilde{P}(s'|s,a) \max_{a'} Q_{upper}(s',a') \quad \forall s,a \tag{6}$$

At convergence, define $V_{upper}(s) = \max_a Q_{upper}(s,a)$. Strehl and Littman (2008) prove that this converges.

Strehl and Littman provide Algorithm UPPERP (Algorithm 2) for solving the optimization over $CI(P(s,a),\delta_1)$ in (6) efficiently. If the radius of the confidence interval is ω, then we can solve for $\tilde{P}$ by shifting $\Delta\omega = \omega/2$ of the probability mass from outcomes s' for which $V_{upper}(s') = \max_{a'} Q_{upper}(s',a')$ is low ("donor states") to outcomes for which it is maximum ("recipient states"). This will result in creating a $\tilde{P}$ distribution that is at L_1 distance ω from $\hat{P}$. The algorithm repeatedly finds a pair of successor states $\underline{s}$ and $\overline{s}$ and shifts probability from one to the other until it has shifted $\Delta\omega$. Note that in most cases, $\overline{s}$ will be a state for which $N(s,a,\overline{s}) = 0$—that is, a state we have never visited. In such cases, $V_{upper}(\overline{s}) = V_{max}$.

As with MBAE, UPPERP only requires time proportional to the number of transitions that have been observed to have non-zero probability.

The MBIE algorithm works as follows. Given the upper bound Q_{upper}, MBIE defines an exploration policy based on the optimism principle (Buşoniu and Munos, 2012). Specifically, at each state s, it selects the action a that maximizes $Q_{upper}(s,a)$. It then performs that action in the MDP simulator to obtain the immediate reward r and the resulting state s'. It then updates its statistics $N(s,a,s')$, $R(s,a)$, and $N(s,a)$ and recomputes Q_{upper}.

MBIE never terminates. However, it does compute a constant m such that if $N(s,a) > m$, then it does not draw a new sample from the MDP simulator for (s,a). Instead, it samples a next state according to its transition probability estimate $\hat{P}(s'|s,a)$. Hence, in an ergodic[2] or unichain[3] MDP, it will eventually stop drawing new samples, because it will have invoked the simulator on all actions a in all non-transient states s at least m times.

2. An ergodic MDP is an MDP where every state can be accessed in a finite number of steps from any other state
3. In unichain MDP, every policy in an MDP result in a single ergodic class

Algorithm 2: UPPERP(s,a,δ,M_0)

Input: s,a
δ: Confidence parameter
M_0: missing mass limit
Lines marked by **GT:** are for the Good-Turing extension
$N(s,a) := \sum_{s'} N(s,a,s')$
$\hat{P}(s'|s,a) := N(s,a,s')/N(s,a)$ for all s'
$\tilde{P}(s'|s,a) := \hat{P}(s'|s,a)$ for all s'
$\Delta\omega := \omega(N(s,a),\delta)/2$
GT: $N_0(s,a) := \{s'|N(s,a,s') = 0\}$
GT: $\Delta\omega := \min\left(\omega(N(s,a),\delta/2)/2, (1+\sqrt{2})\sqrt{\frac{ln(2/\delta)}{N(s,a)}}\right)$
while $\Delta\omega > 0$ **do**
  $S' := \{s' : \hat{P}(s'|s,a) < 1\}$ recipient states
GT: **if** $M_0 = 0$ **then** $S' := S' \setminus N_0(s,a)$
  $\underline{s} := \text{argmin}_{s':\tilde{P}(s'|s,a)>0} V_{upper}(s')$ donor state
  $\overline{s} := \text{argmax}_{s'\in S', \tilde{P}(s'|s,a)<1} V_{upper}(s')$ recipient state
  $\xi := \min\{1-\tilde{P}(\overline{s}|s,a), \tilde{P}(\underline{s}|s,a), \Delta\omega\}$
  $\tilde{P}(\underline{s}|s,a) := \tilde{P}(\underline{s}|s,a) - \xi$
  $\tilde{P}(\overline{s}|s,a) := \tilde{P}(\overline{s}|s,a) + \xi$
  $\Delta\omega := \Delta\omega - \xi$
GT: **if** $\overline{s} \in N_0(s,a)$ **then** $M_0 := M_0 - \xi$
return $\tilde{P}$

Because MBIE does not terminate, it cannot be applied directly to MDP planning. However, we can develop an MDP planning version by using the horizon time H computed by Fiechter's method and forcing MBIE to jump back to s_0 each time it has traveled H steps away from the start state. Algorithm 3 provides the pseudo-code for this variant of MBIE, which we call MBIE-reset.

Now that we have described the application goal and previous research, we present the novel contributions of this paper.

5. Improved Model-Based MDP Planning

We propose a new algorithm, which we call DDV. Algorithm 4 presents the general schema for the algorithm. For each state-action (s,a) pair that has been explored, DDV maintains upper and lower confidence limits on $Q(s,a)$ such that $Q_{lower}(s,a) \leq Q^*(s,a) \leq Q_{upper}(s,a)$ with high probability. From these, we compute a confidence interval on the value of the start state s_0 according to $V_{lower}(s_0) = \max_a Q_{lower}(s_0,a)$ and $V_{upper}(s_0) = \max_a Q_{upper}(s_0,a)$. Consequently, $V_{lower}(s_0) \leq V^*(s_0) \leq V_{upper}(s_0)$ with high probability. The algorithm terminates when the width of this confidence interval, which we denote by $\Delta V(s_0) = V_{upper}(s_0) - V_{lower}(s_0)$, is less than ε.

The confidence intervals for Q_{lower} and Q_{upper} are based on an extension of the Weissman, et al. confidence interval of Equation (5), which we will refer to as $CI^{GT}(P(s,a),\delta_1)$ (which will be described below). The confidence intervals are computed by iterating the following equations to

Algorithm 3: MBIE-reset($s_0, \gamma, F, H, \varepsilon, \delta$)

Input: s_0:start state, γ: discount rate, F: a simulator, H : horizon,ε, δ: accuracy and confidence parameters

$N(s,a,s') = 0$ for all (s,a,s')

$m = c\left[\frac{|S|}{\varepsilon^2(1-\gamma)^4} + \frac{1}{\varepsilon^2(1-\gamma)^4}\ln\frac{|S||A|}{\varepsilon(1-\gamma)\delta}\right]$

repeat forever

  $s = s_0$

  $h = 1$

  while $h \leq H$ **do**

    update Q_{upper} and V_{upper} by iterating equation 6 to convergence

    $a = \text{argmax}_a\, Q_{upper}(s)$

    if $N(s,a) < m$ **then**

      $(r,s') \sim F(s,a)$ // draw sample

      update $N(s,a,s'), N(s,a)$, and $R(s,a)$

    else

      $s' \sim \hat{P}(s'|s,a)$

      $r = R(s,a)$

    $h = h+1$

convergence:

$$Q_{lower}(s,a) = R(s,a) + \min_{\tilde{P}(s,a)\in CI^{GT}(P(s,a),\delta_1)} \gamma \sum_{s'} \tilde{P}(s'|s,a) \max_{a'} Q_{lower}(s',a') \quad \forall s,a. \tag{7}$$

$$Q_{upper}(s,a) = R(s,a) + \max_{\tilde{P}(s,a)\in CI^{GT}(P(s,a),\delta_1)} \gamma \sum_{s'} \tilde{P}(s'|s,a) \max_{a'} Q_{upper}(s',a') \quad \forall s,a. \tag{8}$$

The Q values are initialized as follows: $Q_{lower}(s,a) = 0$ and $Q_{upper}(s,a) = V_{max}$. At convergence, define $V_{lower}(s) = \max_a Q_{lower}(s,a)$ and $V_{upper}(s) = \max_a Q_{upper}(s,a)$.

Lemma 2 *If $\delta_1 = \delta/(|S||A|)$, then with probability $1-\delta$, $Q_{lower}(s,a) \leq Q^*(s,a) \leq Q_{upper}(s,a)$ for all (s,a) and $V_{lower}(s) \leq V^*(s) \leq V_{upper}$ for all s.*

Proof Strehl and Littman (2008) prove this for Q_{upper} and V_{upper} by showing that it is true at the point of initialization and that Equation (8) is a contraction. Hence, it remains true by induction on the number of iterations of value iteration. The proof for Q_{lower} and V_{lower} is analogous. ■

The exploration heuristic for DDV is based on exploring the state-action pair (s,a) that maximizes the expected decrease in $\Delta V(s_0)$. We write this quantity as $\Delta\Delta V(s_0|s,a)$, because it is a change (Δ) in the confidence interval width $\Delta V(s_0|s,a)$. Below, we will describe two different heuristics that are based on two different approximations to $\Delta\Delta V(s_0|s,a)$.

We now present the improved confidence interval, CI^{GT}, and evaluate its effectiveness experimentally. Then we introduce our two search heuristics, analyze them, and present experimental evidence that they improve over previous heuristics.

Algorithm 4: DDV $(s_0, \gamma, F, \varepsilon, \delta)$

Input: s_0:start state
γ: discount rate
F: a simulator
ε, δ: accuracy and confidence parameters
$m = c\left[\frac{|S|}{\varepsilon^2(1-\gamma)^4} + \frac{1}{\varepsilon^2(1-\gamma)^4}\ln\frac{|S||A|}{\varepsilon(1-\gamma)^\delta}\right]$
$\delta' = \delta/(|S||A|m)$
$\tilde{S} = \{s_0\}$ // observed and/or explored states
$N(s,a,s') = 0$ for all (s,a,s')
repeat forever
  update $Q_{upper}, Q_{lower}, V_{upper}, V_{lower}$ by iterating equations 7 and 8 using δ' to compute the confidence intervals
  if $V_{upper}(s_0) - V_{lower}(s_0) \leq \varepsilon$ **then**
    // compute a good policy and terminate
    $\pi_{lower}(s) = \arg\max_a Q_{lower}(s,a)$
    return π_{lower}
  forall the *explored or observed states s* **do**
    forall the *actions a* **do**
      compute $\Delta\Delta V(s_0|s,a)$
  compute $(s,a) := \text{argmax}_{(s,a)}\, \Delta\Delta V(s_0|s,a)$
  $(r,s') \sim F(s,a)$ // draw sample
  $\tilde{S} := \tilde{S} \cup \{s'\}$ // update the set of discovered states
  update $N(s,a,s'), N(s,a)$, and $R(s,a)$

5.1 Tighter Statistical Analysis for Earlier Stopping

The first contribution of this paper is to improve the confidence intervals employed in equation (6). In many real-world MDPs, the transition probability distributions are sparse in the sense that there are only a few states s' such that $P(s'|s,a) > 0$. A drawback of the Weissman et al. confidence interval is that $\omega(N,\delta)$ scales as $O(\sqrt{|S|/N})$, so the intervals are very wide for large state spaces. We would like a tighter confidence interval for sparse distributions.

Our approach is to intersect the Weissman et al. confidence interval with a confidence interval based on the Good-Turing estimate of the missing mass (Good, 1953).

Definition 3 *For a given state-action pair (s,a), let $N_k(s,a) = \{s'|N(s,a,s') = k\}$ be the set of all result states s' that have been observed exactly k times. We seek to bound the total probability of those states that have never been observed: $M_0(s,a) = \sum_{s' \in N_0(s,a)} P(s'|s,a)$. The Good-Turing estimate of $M_0(s,a)$ is*

$$\widehat{M}_0(s,a) = \frac{|N_1(s,a)|}{N(s,a)}.$$

In words, Good and Turing count the number of successor states that have been observed exactly once and divide by the number of samples. The following lemma follows directly from Kearns and Saul (1998), McAllester and Schapire (2000), and McAllester and Ortiz (2003).

Lemma 4 *With probability* $1-\delta$,

$$M_0(s,a) \leq \widehat{M}_0(s,a) + (1+\sqrt{2})\sqrt{\frac{\ln(1/\delta)}{N(s,a)}}. \tag{9}$$

Proof Let $S(M_0(s,a),x)$ be the Chernoff "entropy", defined as

$$S(M_0(s,a),x) = \sup_{\beta} x\beta - \ln Z(M_0(s,a),\beta),$$

where $Z(M_0(s,a),\beta) = \mathbb{E}[e^{\beta M_0(s,a)}]$. McAllester and Ortiz (2003, Theorem 16) prove that

$$S(M_0(s,a),\mathbb{E}[M_0(s,a)]+\varepsilon) \geq N(s,a)\varepsilon^2.$$

From Lemmas 12 and 13 of McAllester and Schapire (2000),

$$\mathbb{E}[M_0(s,a)] \leq \widehat{M}_0(s,a) + \sqrt{\frac{2\log 1/\delta}{N(s,a)}}.$$

Combining these results yields

$$S\left(M_0(s,a),\widehat{M}_0(s,a) + \sqrt{\frac{2\log 1/\delta}{N(s,a)}} + \varepsilon\right) \geq N(s,a)\varepsilon^2. \tag{10}$$

Chernoff (1952) proves that

$$P(M_0(s,a) \geq x) \leq e^{-S(M_0(s,a),x)}.$$

Plugging in (10) gives

$$P\left(M_0(s,a) \geq \widehat{M}_0(s,a) + \sqrt{\frac{2\log 1/\delta}{N(s,a)}} + \varepsilon\right) \leq e^{-N(s,a)\varepsilon^2}. \tag{11}$$

Setting $\delta = e^{-N(s,a)\varepsilon^2}$ and solving for ε gives $\varepsilon = \sqrt{(\log 1/\delta)/N(s,a)}$. Plugging this into (11) and simplifying gives the result. ■

Define $CI^{GT}(\hat{P}|N(s,a),\delta)$ to be the set of all distributions $\tilde{P} \in CI(\hat{P}|N(s,a),\delta/2)$ such that $\sum_{s' \in N_0(s,a)} \tilde{P}(s'|s,a) < \widehat{M}_0(s,a) + (1+\sqrt{2})\sqrt{\frac{\ln(2/\delta)}{N(s,a)}}$. This intersects the Weissman and Good-Turing intervals. Note that since we are intersecting two confidence intervals, we must compute both (5) and (9) using $\delta/2$ so that they will simultaneously hold with probability $1-\delta$.

We can incorporate the bound from (9) into UPPERP by adding the lines prefixed by "GT:" in Algorithm 2. These limit the amount of probability that can be shifted to unobserved states according to (9). The modified algorithm still only requires time proportional to the number of states s' where $N(s,a,s') > 0$.

5.1.1 EXPERIMENTAL EVALUATION OF THE IMPROVED CONFIDENCE BOUND

To test the effectiveness of this Good-Turing improvement, we ran MBIE-reset and compared its performance with and without the improved confidence interval.

We experimented with four MDPs. The first is a Combination Lock MDP with 500 states. In each state i, there are two possible actions. The first action makes a deterministic transition to state $i+1$ with reward 0 except for state 500, which is a terminal state with a reward of 1. The second action makes a transition (uniformly) to one of the states $1,\ldots,i-1$ with reward 0. The optimal policy is to choose the first action in every state, even though it doesn't provide a reward until the final state.

The remaining three MDPs are different versions of the tamarisk management MDP. The specific network configurations that we employed in this experiment were the following:

- $E = 3, H = 2$ with the budget constraint that in each time step we can only choose one edge in which to perform a non-"do nothing" action. This gives a total of 7 actions.
- $E = 3, H = 3$ with the same constraints as for $E = 3, H = 2$.
- $E = 7, H = 1$ with the budget constraint that in each time step we can only choose one edge in which to perform a non-"do nothing" action. The only such action is "restore". This gives a total of 8 actions.

5.1.2 RESULTS

Figure 2 shows the upper and lower confidence bounds, $V_{upper}(s_0)$ and $V_{lower}(s_0)$, on the value of the starting state s_0 as a function of the number of simulator calls. The confidence bounds for the Weissman et al. interval are labeled "V(CI)", whereas the bounds for this interval combined with the Good-Turing interval are labeled "V(CI-GT)".

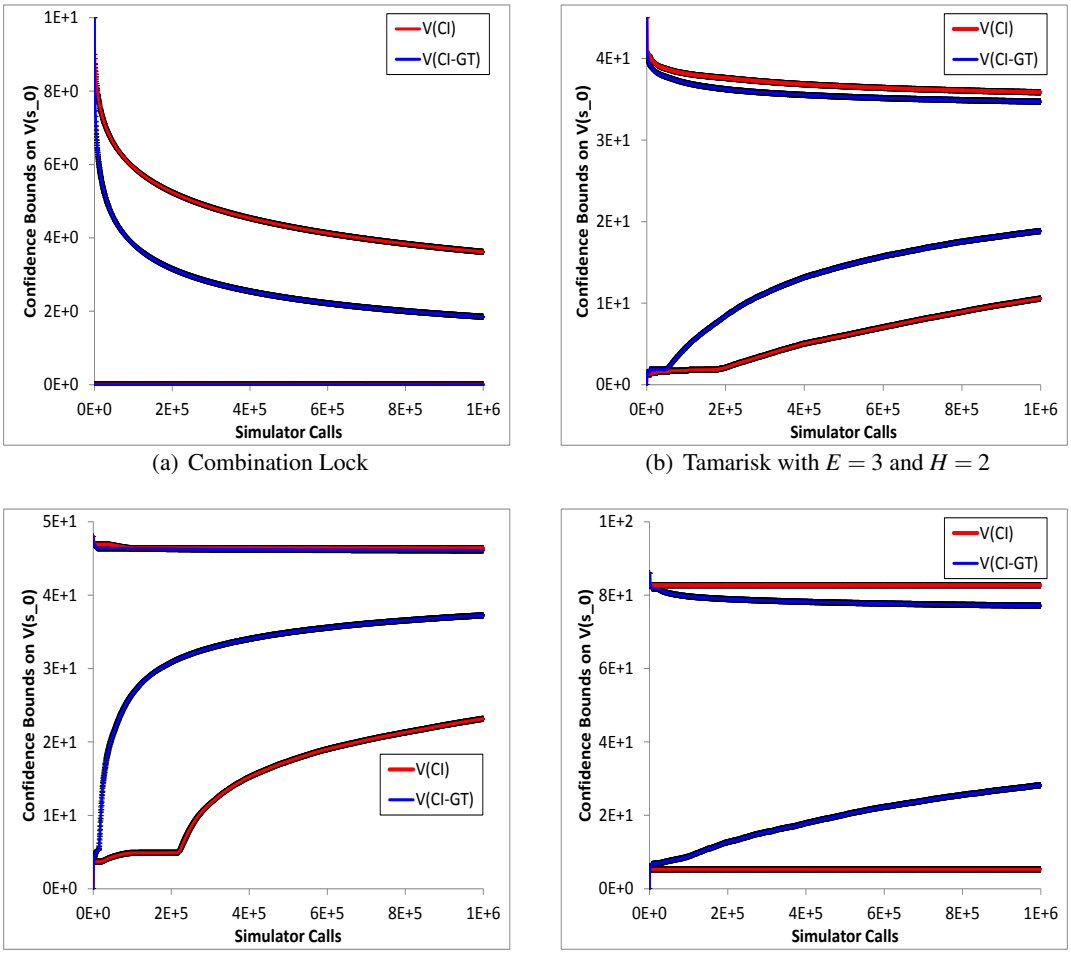
5.1.3 DISCUSSION

The results show that the Good-Turing interval provides a substantial reduction in the number of required simulator calls. On the combination lock problem, the CI-GT interval after 2×10^5 calls is already better than the CI interval after 10^6 calls, for a more than five-fold speedup. On the $E = 3, H = 2$ tamarisk problem, the speedup is more than a factor of three. On the $E = 3, H = 3$ version, the speedup is more than five-fold. And on the $E = 7, H = 1$ problem, the CI interval does not show any progress toward convergence, whereas the CI-GT interval has begun to make progress.

5.2 Improved Exploration Heuristics for MDP Planning

The second contribution of this paper is to define two new exploration heuristics for MDP planning and compare them to existing algorithms. As with previous work, we wish to exploit reachability and discounting to avoid exploring unnecessarily. However, we want to take advantage of the fact that our simulators are "strong" in the sense that we can explore any desired state-action pair in any order.

As discussed above, our termination condition is to stop when the width of the confidence interval $\Delta V(s_0) = V_{upper}(s_0) - V_{lower}(s_0)$ is less than ε. Our heuristics are based on computing the state-action pair (s,a) that will lead to the largest (one step) reduction in $\Delta V(s_0)$. Formally, let



(a) Combination Lock

(b) Tamarisk with $E = 3$ and $H = 2$

(c) Tamarisk with $E = 3$ and $H = 3$

(d) Tamarisk with $E = 7$ and $H = 1$

Figure 2: Plots of $V_{upper}(s_0)$ and $V_{lower}(s_0)$ for MBIE-reset on $V(s_0)$ with and without incorporating Good-Turing confidence intervals. Values are the mean of 15 independent trials. Error bars (which are barely visible) show 95% confidence intervals computed from the 15 trials.

$\Delta\Delta V(s_0|s,a) = \mathbb{E}[\Delta V(s_0) - \Delta V'(s_0)|(s,a)]$ be the expected change in $\Delta V(s_0)$ if we draw one more sample from (s,a). Here the prime in $\Delta V'(s_0)$ denotes the value of $\Delta V(s_0)$ after exploring (s,a). The expectation is taken with respect to two sources of uncertainty: uncertainty about the reward $R(s,a)$ and uncertainty about the resulting state $s' \sim P(s'|s,a)$.

Suppose we are considering exploring (s,a). We approximate $\Delta\Delta V(s_0|s,a)$ in two steps. First, we consider the reduction in our uncertainty about $Q(s,a)$ if we explore (s,a). Let $\Delta Q(s,a) = Q_{upper}(s,a) - Q_{lower}(s,a)$ and $\Delta\Delta Q(s,a) = \mathbb{E}[\Delta Q(s,a) - \Delta Q'(s,a)|(s,a)]$. Second, we consider the impact that reducing $\Delta Q(s,a)$ will have on $\Delta V(s_0)$.

We compute $\Delta\Delta Q(s,a)$ as follows.

Case 1: $N(s,a) = 0$. In this case, our current bounds are $Q_{lower}(s,a) = 0$ and $Q_{upper}(s,a) = V_{max}$. After we sample $(r,s') \sim F(s,a)$, we will observe the actual reward $R(s,a) = r$ and we will observe one of the possible successor states s'. For purposes of deriving our heuristic, we will assume a uniform[4] prior on $R(s,a)$ so that the expected value of R is $\overline{R} = R_{max}/2$. We will assume that s' will be a "new" state that we have never observed before, and hence $V_{upper}(s') = V_{max}$ and $V_{lower}(s') = 0$. This gives us

$$Q'_{upper}(s,a) = \overline{R}(s,a) + \gamma R_{max}/(1-\gamma) \tag{12a}$$

$$Q'_{lower}(s,a) = \overline{R}(s,a), \tag{12b}$$

If a more informed prior is known for $R(s,a)$, then it could be employed to derive a more informed exploration heuristic.

Case 2: $N(s,a) > 0$. In this case, we have already observed $R(s,a)$, so it is no longer a random variable. Hence, the expectation is only over s'. For purposes of deriving our exploration heuristic, we will assume that s' will be drawn according to our current maximum likelihood estimate $\hat{P}(s'|s,a)$ but that $N_1(s,a)$ will not change. Consequently, the Good-Turing estimate will not change. Under this assumption, the expected value of Q will not change, $M_0(s,a)$ will not change, so the only change to Q_{upper} and Q_{lower} will result from replacing $\omega(N(s,a),\delta)$ by $\omega(N(s,a)+1,\delta)$ in the Weissman et al. confidence interval.

Note that DDV may explore a state-action pair (s,a) even if a is not currently the optimal action in s. That is, even if $Q_{upper}(s,a) < Q_{upper}(s,a')$ for some $a' \neq a$. An alternative rule would be to only explore (s,a) if it would reduce the expected value of $\Delta V(s) = V_{upper}(s) - V_{lower}(s)$. However, if there are two actions a and a' such that $Q_{upper}(s,a) = Q_{upper}(s,a')$, then exploring only one of them will not change $\Delta V(s)$. Our heuristic avoids this problem. We have studied another variant in which we defined $V_{upper}(s) = \text{softmax}_a(\tau)\, Q_{upper}(s,a)$ (the softmax with temperature τ). This gave slightly better results, but it requires that we tune τ, which is a nuisance.

The second component of our heuristic is to estimate the impact of $\Delta\Delta Q(s_0|s,a)$ on $\Delta\Delta V(s_0|s,a)$. To do this, we appeal to the concept of an occupancy measure.

Definition 5 *The* occupancy measure $\mu^\pi(s)$ *is the expected discounted number of times that policy* π *visits state s,*

$$\mu^\pi(s) = \mathbb{E}\left[\sum_{t=1}^{\infty} \gamma^t \mathbb{I}[s_t = s] \;\middle|\; s_0, \pi\right], \tag{13}$$

where $\mathbb{I}[\cdot]$ *is the indicator function and the expectation taken is with respect to the transition distribution.*

This can be computed via dynamic programming on the Bellman flow equation (Syed et al., 2008):

$$\mu^\pi(s) := P_0(s) + \gamma \sum_{s^-} \mu^\pi(s^-) P(s|s^-, \pi(s^-)).$$

This says that the (discounted) probability of visiting state s is equal to the sum of the probability of starting in state s (as specified by the starting state distribution $P_0(s)$) and the probability of reaching s by first visiting state s^- and then executing an action that leads to state s.

4. Any symmetric prior centered on $R_{max}/2$ would suffice.

The intuition behind using an occupancy measure is that if we knew that the optimal policy would visit s with measure $\mu^*(s)$ and if exploring (s,a) would reduce our uncertainty at state s by approximately $\Delta\Delta Q(s_0|s,a)$, then a reasonable estimate of the impact on $\Delta V(s_0)$ would be $\mu^*(s)\Delta\Delta Q(s_0|s,a)$. Unfortunately, we don't know μ^* because we don't know the optimal policy. We consider two other occupancy measures instead: μ^{OUU} and $\overline{\mu}$.

The first measure, μ^{OUU} is computed based on the principle of optimism under uncertainty. Specifically, define $\pi^{OUU}(s) := \max_a Q_{upper}(s,a)$ to be the policy that chooses the action that maximizes the upper confidence bound on the Q function. This is the policy followed by MBIE and most other upper-confidence bound methods. This gives us the DDV-OUU heuristic.

Definition 6 *The DDV-OUU heuristic explores the state action pair (s,a) that maximizes*

$$\mu^{OUU}(s)\Delta\Delta Q(s_0|s,a).$$

The second measure $\overline{\mu}$ is computed based on an upper bound of the occupancy measure over all possible policies. It gives us the DDV-UPPER heuristic.

Definition 7 *The DDV-UPPER heuristic explores the state action pair (s,a) that maximizes*

$$\overline{\mu}(s)\Delta\Delta Q(s_0|s,a).$$

The next section defines $\overline{\mu}$ and proves a property that may be of independent interest.

5.2.1 AN UPPER BOUND ON THE OCCUPANCY MEASURE

The purpose of this section is to introduce $\overline{\mu}$, which is an upper bound on the occupancy measure of any optimal policy for a restricted set of MDPs $\widetilde{\mathscr{M}}$. This section defines this measure and presents a dynamic programming algorithm to compute it. The attractive aspect of $\overline{\mu}$ is that it can be computed without knowing the optimal policy. In this sense, it is analogous to the value function, which value iteration computes in a policy-independent way.

To define $\overline{\mu}$, we must first define the set $\widetilde{\mathscr{M}}$ of MDPs. At each point during the execution of DDV, the states S of the unknown MDP can be partitioned into three sets: (a) the *unobserved states* s (i.e., $N(s^-,a^-,s)=0$ for all s^-,a^-); (b) the *observed but unexplored states* s (i.e., $\exists(s^-,a^-)N(s^-,a^-,s)>0$ but $N(s,a)=0$ for all a), and (c) *the (partially) explored states* s (i.e., $N(s,a,s')>0$ for some a). Consider the set $\widetilde{\mathscr{M}} = \langle\tilde{S},\tilde{A},\tilde{T},\tilde{R},s_0\rangle$ of MDPs satisfying the following properties:

- $\tilde{S}$ consists of all states s that have been either observed or explored,
- $\tilde{A}=A$, the set of actions in the unknown MDP,
- $\tilde{T}$ consists of any transition function T such that for explored states s and all actions a, $T(s,a,\cdot)\in CI^{GT}(\hat{P}(s,a),\delta)$. For all observed but not explored states s, $T(s,a,s)=1$ for all a, so they enter self-loops.
- $\tilde{R}$: For explored (s,a) pairs, $\tilde{R}(s,a)=R(s,a)$. For unexplored (s,a) pairs, $\tilde{R}(s,a)\in[0,R_{max}]$.
- s_0 is the artificial start state.

The set $\widetilde{\mathscr{M}}$ contains all MDPs consistent with the observations with the following restrictions. First, the MDPs do not contain any of the unobserved states. Second, the unexplored states contain self-loops and hence do not transition to any other states.

Define $P_{upper}(s'|s,a)$ as follows:

$$P_{upper}(s'|s,a) = \max_{\tilde{P}(s,a) \in CI^{GT}(P,\delta)} \tilde{P}(s'|s,a).$$

Define $\overline{\mu}$ as the solution to the following dynamic program. For all states s,

$$\overline{\mu}(s) = \sum_{s^- \in pred(s)} \max_{a^-} \gamma P_{upper}(s|s^-,a^-)\overline{\mu}(s^-). \tag{14}$$

The intuition is that we allow each predecessor s^- of s to choose the action a^- that would send the most probability mass to s and hence give the biggest value of $\overline{\mu}(s)$. These action choices a^- are not required to be consistent for multiple successors of s^-. We fix $\overline{\mu}(s_0) = \mu(s_0) = 1$. (Recall, that s_0 is an artificial start state. It is not reachable from any other state—including itself—so $\mu(s_0) = 1$ for all policies.)

Lemma 8 *For all MDPs $\widetilde{M} \in \widetilde{\mathscr{M}}$, $\overline{\mu}(s) \geq \mu^{\pi^*(\widetilde{M})}(s)$, where $\pi^*(\widetilde{M})$ is any optimal policy of $\widetilde{M}$.*

Proof By construction, $P_{upper}(s'|s,a)$ is the maximum over all transition distributions in $\widetilde{\mathscr{M}}$ of the probability of $(s,a) \to s'$. According to (14), the probability flowing to s is the maximum possible over all policies executed in the predecessor states $\{s^-\}$. Finally, all probability reaching a state s must come from its known predecessors $pred(s)$, because all observed but unexplored states only have self-transitions and hence cannot reach s or any of its predecessors. ■

In earlier work, Smith and Simmons (2006) employed a less general path-specific bound on μ as a heuristic for focusing Real-Time Dynamic Programming (a method that assumes a full model of the MDP is available).

5.2.2 SOUNDNESS OF DDV-OUU AND DDV-UPPER

We now show that DDV, using either of these heuristics, produces an ε-optimal policy with probability at least $1-\delta$ after making only polynomially-many simulator calls. The steps in this proof closely follow previous proofs by Strehl and Littman (2008) and Even-Dar et al. (2006).

Theorem 9 (DDV is PAC-RL) *There exists a sample size m polynomial in $|S|$, $|A|$, $1/\varepsilon$, $1/\delta$, $1/(1-\gamma)$, R_{max}, such that* $\mathrm{DDV}(s_0,F,\varepsilon,\delta/(m|S||A|))$ *with either the DDV-OUU or the DDV-UPPER heuristic terminates after no more than $m|S||A|$ calls on the simulator and returns a policy π such that $|V^{\pi}(s_0) - V^*(s_0)| < \varepsilon$ with probability $1-\delta$.*

Proof First, note that every sample drawn by DDV will shrink the confidence interval for some $Q(s,a)$. Hence, these intervals will eventually become tight enough to make the termination condition true. To establish a rough bound on sample complexity, let us suppose that each state must be sampled enough so that $\Delta Q(s,a) = Q_{upper}(s,a) - Q_{lower}(s,a) \leq \varepsilon$.

This will cause termination. Consider state s_0 and let $a_{upper} = \mathrm{argmax}_a Q_{upper}(s_0,a)$ be the action chosen by the OUU policy. Then the upper bound on s is $V_{upper}(s) = Q_{upper}(s,a_{upper})$, and

the lower bound on S is $V_{lower}(s_0) = \max_a Q_{lower}(s_0,a) \geq Q_{lower}(s_0,a_{upper})$. Hence, the difference $V_{upper}(s_0) - V_{lower}(s_0) \leq \varepsilon$.

How many samples are required to ensure that $\Delta Q(s,a) \leq \varepsilon$ for all (s,a)? We can bound $Q_{upper}(s,a) - Q_{lower}(s,a)$ as follows.

$$\begin{aligned} Q_{upper}(s,a) - Q_{lower}(s,a) &= R(s,a) + \gamma \max_{\tilde{P} \in CI(\hat{P}(s,a),\delta')} \sum_{s'} \tilde{P}(s'|s,a) V_{upper}(s') \\ &\quad -R(s,a) - \gamma \min_{\tilde{P} \in CI(\hat{P}(s,a),\delta')} \sum_{s'} \tilde{P}(s'|s,a) V_{lower}(s') \end{aligned}$$

Let P_{upper} be the $\tilde{P}$ chosen in the max and P_{lower} be the $\tilde{P}$ chosen in the min. At termination, we know that in every state $V_{upper} \leq V_{lower} + \varepsilon$. Substituting these and simplifying gives

$$Q_{upper}(s,a) - Q_{lower}(s,a) \leq \gamma \sum_{s'} [P_{upper}(s'|s,a) - P_{lower}(s'|s,a)] V_{lower}(s') + \gamma\varepsilon.$$

We make two approximations: $P_{upper}(s'|s,a) - P_{lower}(s'|s,a) \leq |P_{upper}(s'|s,a) - P_{lower}(s'|s,a)|$ and $V_{lower}(s') \leq \frac{R_{max}}{1-\gamma}$. This yields

$$Q_{upper}(s,a) - Q_{lower}(s,a) \leq \gamma \frac{R_{max}}{1-\gamma} \sum_{s'} |P_{upper}(s'|s,a) - P_{lower}(s'|s,a)| + \gamma\varepsilon.$$

We know that $\|P_{upper}(\cdot|s,a) - P_{lower}(\cdot|s,a)\|_1 \leq 2\omega$, because both distributions belong to the L_1 ball of radius ω around the maximum likelihood estimate $\hat{P}$.

$$Q_{upper}(s,a) - Q_{lower}(s,a) \leq \gamma \frac{R_{max}}{1-\gamma} 2\omega + \gamma\varepsilon.$$

Setting this less than or equal to ε and solving for ω gives

$$\omega \leq \frac{\varepsilon(1-\gamma)^2}{2\gamma R_{max}}.$$

We know that

$$\omega = \sqrt{\frac{2[\ln(2^{|S|} - 2) - \ln \delta']}{N}}.$$

To set δ', we must divide δ by the maximum number of confidence intervals computed by the algorithm. This will be $2|S||A|N$, because we compute two intervals (upper and lower) for ever (s,a). Plugging the value for δ' in and simplifying gives the following equation:

$$N \geq \frac{\gamma^2 8 R_{max}^2 [\ln(2^{|S|} - 2) - \ln \delta + \ln 2|S||A| + \ln N]}{\varepsilon^2 (1-\gamma)^4}.$$

This has no closed form solution. However, as Strehl and Littman note, there exists a constant C such that if $N \geq 2C \ln C$ then $N \geq C \ln N$. Hence, the $\ln N$ term on the right-hand side will only require a small increase in N. Hence

$$N = O\left(\frac{\gamma^2 R_{max}^2 |S| + \ln |S||A|/\delta}{\varepsilon^2 (1-\gamma)^4} \right).$$

In the worst case, we must draw N samples for every state-action pair, so

$$m = O\left(|S|^2|A|\frac{\gamma^2 R_{max}^2 + \ln|S||A|/\delta}{\varepsilon^2(1-\gamma)^4}\right),$$

which is polynomial in all of the relevant parameters.

To prove that the policy output by DDV is within ε of optimal with probability $1-\delta$, note that the following relationships hold:

$$V_{upper}(s_0) \geq V^*(s_0) \geq V^{\pi_{lower}}(s_0) \geq V_{lower}(s_0).$$

The inequalities $V_{upper}(s_0) \geq V^*(s_0) \geq V_{lower}(s_0)$ hold (with probability $1-\delta$) by the admissibility of the confidence intervals. The inequality $V^*(s_0) \geq V^{\pi_{lower}}(s_0)$ holds, because the true value of any policy is no larger than the value of the optimal policy. The last inequality, $V^{\pi_{lower}}(s_0) \geq V_{lower}(s_0)$, holds because extended value iteration estimates the value of π_{lower} by backing up the values V_{lower} of the successor states. At termination, $V_{upper}(s_0) - V_{lower}(s_0) \leq \varepsilon$. Hence, $V^*(s_0) - V^{\pi_{lower}}(s_0) \leq \varepsilon$. ■

5.3 Experimental Evaluation on Exploration Heuristics

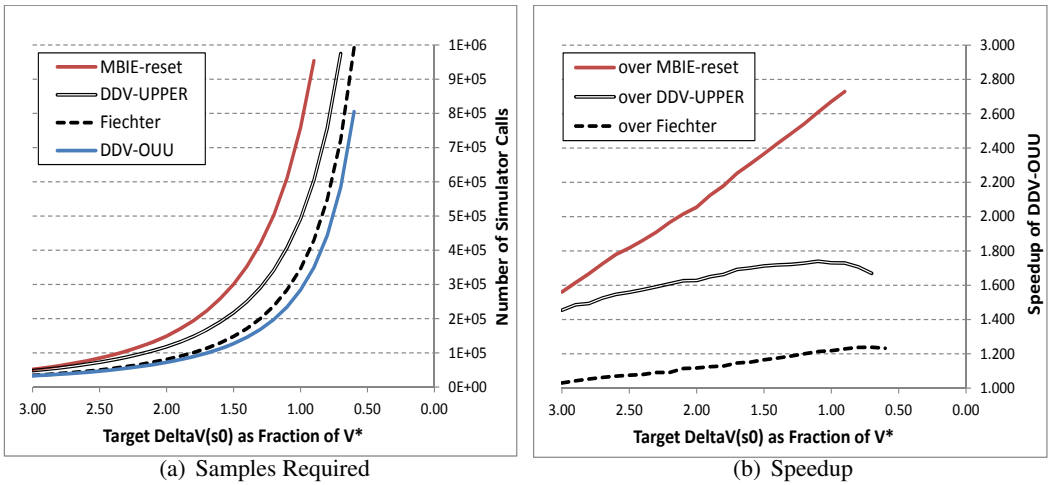
We conducted an experimental study to assess the effectiveness of DDV-OUU and DDV-UPPER and compare them to the exploration heuristics of MBIE (with reset) and Fiechter's algorithm.

5.3.1 METHODS

We conducted two experiments. The goal of both experiments was to compare the number of simulator calls required by each algorithm to achieve a target value ε for the width of the confidence interval, $\Delta V(s_0)$, on the value of the optimal policy in the starting state s_0. For problems where the value $V^*(s_0)$ of the optimal policy is known, we define $\varepsilon = \alpha V^*(s_0)$ and plot the required sample size as a function of α. For the tamarisk problems, where $V^*(s_0)$ is not known, we define $\varepsilon = \alpha R_{max}$ and again plot the required sample size as a function of α. This is a natural way for the user to define the required accuracy ε.

In the first experiment, we employed four MDPs: the RiverSwim and SixArms benchmarks, which have been studied by Strehl and Littman (2004, 2008), and two instances of our tamarisk management MDPs ($E = 3$, $H = 1$) and ($E = 3$, $H = 2$). Each of the tamarisk MDPs implemented a budget constraint that permits a non-"do nothing" action in only one edge in each time step. In the $E = 3, H = 2$ MDP, we included exogenous arrivals using the parameters described in Section 3 (up to 10 seeds per species per edge; Bernoulli parameters are 0.1 for tamarisk and 0.4 for native plants). The $E = 3, H = 1$ tamarisk MDP has 7 actions and 27 states, and the $E = 3, H = 2$ MDP has 7 actions and 216 states. The discount factor was set to 0.9 in all four MDPs.

Each algorithm was executed for one million simulator calls. Instead of performing dynamic programming updates (for extended value iteration and occupancy measure computation) after every simulator call, we computed them on the following schedule. For MBIE-reset, we performed dynamic programming after each complete trajectory. For DDV-OUU and DDV-UPPER, we performed dynamic programming after every 10 simulator calls. Extended value iteration gives us the confidence limits $V_{lower}(s_0)$ and $V_{upper}(s_0)$ for the starting state from which we also computed



(a) Samples Required (b) Speedup

Figure 3: RiverSwim results: (a) Number of samples required by MBIE-reset, Fiechter, DDV-UPPER, and DDV-OUU to achieve various target confidence interval widths $\Delta V(s_0)$. (b) Speedup of DDV-OUU over the algorithms.

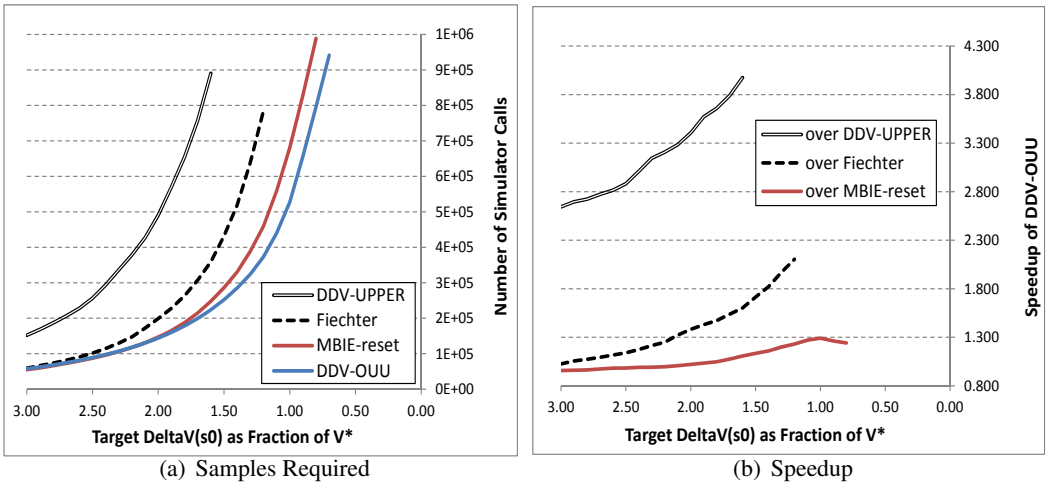
$\Delta V(s_0) = V_{upper}(s_0) - V_{lower}(s_0)$. The experiment was repeated 15 times, and the average value of $\Delta V(s_0)$ was computed. For each MDP, we defined a range of target values for $\Delta V(s_0)$ and computed the average number of samples m required by each algorithm to achieve each target value. By plotting these values, we can see how the sample size increases as we seek smaller target values for $\Delta V(s_0)$. We also computed the speedup of DDV-OUU over each of the other algorithms, according to the formula $m_{alg}/m_{\text{DDV-OUU}}$, and plotted the result for each MDP.

We also measured the total amount of CPU time required by each algorithm to complete the one million simulator calls. Because the simulators in these four MDPs are very efficient, the CPU time primarily measures the cost of the various dynamic programming computations. For Fiechter, these involve setting up and solving the exploration MDP. For MBIE-reset, the primary cost is performing extended value iteration to update V_{upper} and π^{OUU}. For the DDV methods, the cost involves extended value iteration for both V_{upper} and V_{lower} as well as the dynamic program for μ.

In the second experiment, we ran all four algorithms on the RiverSwim and SixArms problems until either 40 million calls had been made to the simulator or until $\Delta V(s_0) \leq \alpha R_{max}$, where $\alpha = 0.1$ and $R_{max} = 10000$ (for RiverSwim) and $R_{max} = 6000$ (for SixArms).

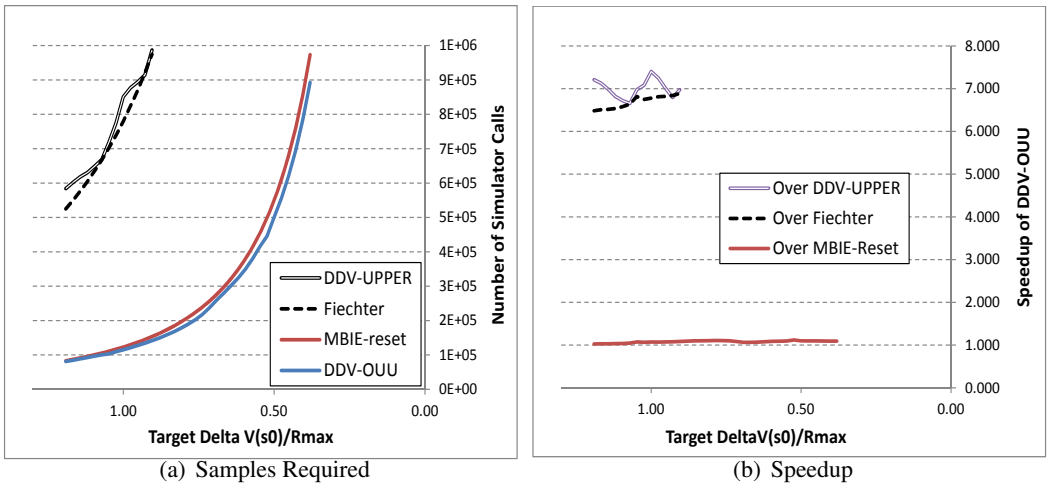
5.3.2 RESULTS

Figures 3, 4, 5, and 6 show the results for the first experiment. In each figure, the left plot shows how the required sample size increases as the target width for $\Delta V(s_0)$ is made smaller. In each figure, the right plot shows the corresponding speedup of DDV-OUU over each of the other algorithms. In all cases, DDV-OUU generally requires the fewest number of samples to reach the target width, and DDV-UPPER generally requires the most. The poor behavior of DDV-UPPER suggests that the policy-free occupancy measure $\overline{\mu}$ is too loose to provide a competitive heuristic.



(a) Samples Required (b) Speedup

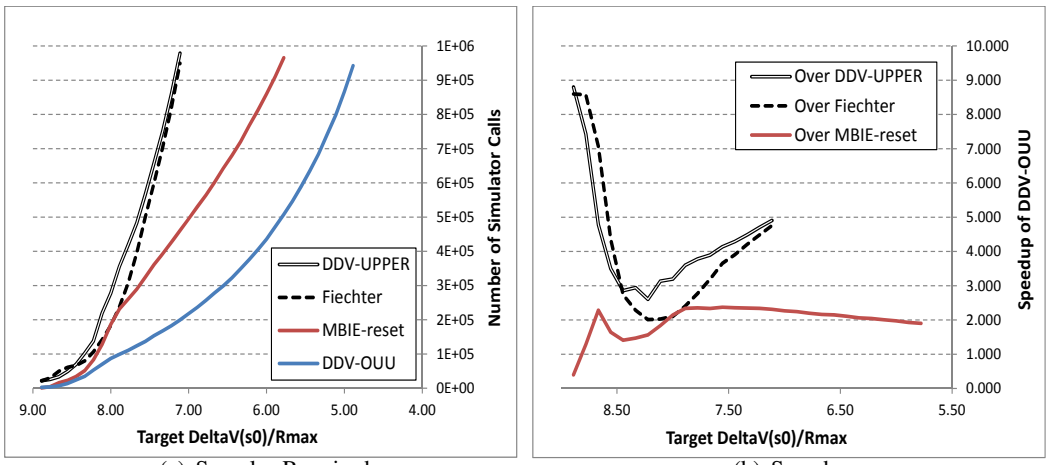
Figure 4: SixArms results: (a) Number of samples required by MBIE-reset, Fiechter, DDV-UPPER, and DDV-OUU to achieve various target confidence interval widths $\Delta V(s_0)$. (b) Speedup of DDV-OUU over the other algorithms.



(a) Samples Required (b) Speedup

Figure 5: Tamarisk with $E = 3$ and $H = 1$ results: (a) Number of samples required by MBIE-reset, Fiechter, DDV-UPPER, and DDV-OUU to achieve various target confidence interval widths $\Delta V(s_0)$. (b) Speedup of DDV-OUU over the other algorithms.

The relative performance of MBIE-reset and Fiechter's algorithm varies dramatically across the four MDPs. On RiverSwim, Fiechter's method is almost as good as DDV-OUU: DDV-OUU shows a speedup of at most 1.23 (23%) over Fiechter. In contrast, MBIE-reset performs much worse. But on SixArms, it is MBIE-reset that is the closest competitor to DDV-OUU. In fact, MBIE-reset is



(a) Samples Required (b) Speedup

Figure 6: Tamarisk with $E = 3$ and $H = 2$ results: (a) Number of samples required by MBIE-reset, Fiechter, DDV-UPPER, and DDV-OUU to achieve various target confidence interval widths $\Delta V(s_0)$. (b) Speedup of DDV-OUU over the other algorithms.

MDP	Algorithm			
	DDV-UPPER (ms/call)	DDV-OUU (ms/call)	MBIE-reset (ms/call)	Fiechter (ms/call)
RiverSwim	9.59	9.92	3.73	3.29
SixArms	15.54	48.97	10.53	4.87
Tamarisk (E=3 and H=1)	11.93	8.13	4.81	4.68
Tamarisk (E=3 and H=2)	187.30	166.79	12.63	18.79

Table 1: RiverSwim clock time per simulator call.

Quantity	Algorithm				
	DDV-UPPER	DDV-OUU	MBIE-reset	Fiechter	Optimal
$V_{upper}(s_0)$	2967.2	2936.6	3001.5	2952.6	2203
$V_{lower}(s_0)$	1967.2	1936.6	2001.5	1952.6	2203
$\Delta V(s_0)$	1000	1000	1000	1000	
Simulator Calls ($\times 10^6$)	2.31	**1.44**	4.05	1.76	

Table 2: RiverSwim confidence intervals and required sample size to achieve target $\Delta V(s_0) = 1000$.

Quantity	Algorithm				
	DDV-UPPER	DDV-OUU	MBIE-reset	Fiechter	Optimal
$V_{upper}(s_0)$	5576.7	5203.9	5242.4	5672.8	4954
$V_{lower}(s_0)$	4140.4	4603.9	4642.4	3997.7	4954
$\Delta V(s_0)$	1436.3	600	600	1675.1	
Simulator Calls ($\times 10^6$)	40.0	**14.5**	19.3	40.0	

Table 3: SixArms confidence intervals and required sample size to achieve the target $\Delta V(s_0) = 600$.

actually better than DDV-OUU for target values larger than 2.1, but as the target width for $\Delta V(s_0)$ is made smaller, DDV-OUU scales much better. On the tamarisk $R = 3$ $H = 1$ problem, MBIE-reset is again almost as good as DDV-OUU. The maximum speedup produced by DDV-OUU is 1.11. Finally, on the tamarisk $R = 3$ $H = 2$ problem, DDV-OUU is definitely superior to MBIE-reset and achieves speedups in the 1.9 to 2.3 range. Surprisingly, on this problem, Fiechter's method is sometimes worse than DDV-UPPER.

The CPU time consumed per simulator call by each algorithm on each problem is reported in Table 1. Not surprisingly, MBIE-reset and Fiechter have much lower cost than the DDV methods. All of these methods are designed for problems where the simulator is extremely expensive. For example, in the work of Houtman et al. (2013) on wildfire management, one call to the simulator can take several minutes. In such problems, the overhead of complex algorithms such as DDV more than pays for itself by reducing the number of simulator calls.

Tables 2 and 3 report the results of the second experiment. The results are consistent with those of the first experiment. DDV-OUU reaches the target $\Delta V(s_0)$ with the smallest number of simulator calls on both problems. On RiverSwim, Fiechter's method is second best, whereas on SixArms, MBIE-reset is second best. On SixArms, DDV-UPPER and Fiechter did not reach the target accuracy within the limit of 40 million simulator calls.

5.3.3 DISCUSSION

The experiments show that DDV-OUU is the most effective of the four algorithms and that it achieves substantial speedups over the other three algorithms (maximum speedups of 2.73 to 7.42 across the four problems).

These results contrast with our previous work (Dietterich, Alkaee Taleghan, and Crowley, 2013) in which we showed that DDV-UPPER is better than MBIE. The key difference is that in the present paper, we are comparing against MBIE with reset, whereas in the previous work, we compared against MBIE without reset. Without resetting, MBIE can spend most of its time in regions of the MDP that are far from the start state, so it can fail to find a good policy for s_0. This behavior also explains the poor performance of Q-learning reported in Dietterich et al. (2013).

6. Summary and Conclusions

This paper has addressed the problem of MDP planning when the MDP is defined by an expensive simulator. In this setting, the planning phase is separate from the execution phase, so there is no

tradeoff between exploration and exploitation. Instead, the goal is to compute a PAC-optimal policy while minimizing the number of calls to the simulator. The policy is designed to optimize the cumulative discounted reward starting in the current real-world state s_0. Unlike in most published RL papers, which typically assume that the MDP is ergodic, the starting state of our ecosystem management problems is typically a transient state.

The paper makes two contributions. First, it shows how to combine the Good-Turing estimate with the L_1-confidence region of Weissman et al. (2003) to obtain tighter confidence intervals (and hence, earlier termination) in sparse MDPs. Second, it shows how to use occupancy measures to create better exploration heuristics. The paper introduced a new policy-independent upper bound $\overline{\mu}$ on the occupancy measure of the optimal policy and applied this to define the DDV-UPPER algorithm. The paper also employed an occupancy measure μ^{OUU} based on the "optimism under uncertainty" principle to define the DDV-OUU algorithm.

The $\overline{\mu}$ measure is potentially of independent interest. Like the value function computed during value iteration, it does not quantify the behavior of any particular policy. This means that it can be computed without needing to have a specific policy to evaluate. However, the DDV-UPPER exploration heuristic did not perform very well. We have two possible explanations for this. First, $\overline{\mu}$ can be a very loose upper bound on the optimal occupancy measure μ^*. Perhaps this leads DDV-UPPER to place too much weight on unfruitful state-action pairs. Second, it is possible that while DDV-UPPER is optimizing the one-step gain in $\Delta\Delta V(s_0)$ (as it is designed to do), DDV-OUU does a better job of optimizing gains over the longer term. Further experimentation is needed to determine which of these explanations is correct.

Our DDV-OUU method gave the best performance in all of our experiments. This is yet another confirmation of the power of the "Optimism Principle" (Buşoniu and Munos, 2012) in exploration. Hence, we recommend it for solving simulator-defined MDP planning problems. We are applying it to solve moderate-sized instances of our tamarisk MDPs. However, additional algorithm innovations will be required to solve much larger tamarisk instances.

Three promising directions for future research are (a) exploiting tighter confidence interval methods, such as the Empirical Bernstein Bound (Audibert et al., 2009; Szita and Szepesvári, 2010) or improvements on the Good-Turing estimate (Orlitsky et al., 2003; Valiant and Valiant, 2013), (b) explicitly formulating the MDP planning problem in terms of sequential inference (Wald, 1945), which would remove the independence assumption in the union bound for partitioning δ, and (c) studying exploration methods based on posterior sampling (Thompson, 1933).

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grants 0832804 and 1331932.

References

Jean Yves Audibert, Remi Munos, and Csaba Szepesvári. Exploration-exploitation trade-off using variance estimates in multi-armed bandits. *Theoretical Computer Science*, 410(19):1876–1902, 2009.

Mohammad Gheshlaghi Azar, Remi Munos, and Hilbert J Kappen. On the sample complexity of reinforcement learning with a generative model. In *Proceedings of the 29th International*

Conference on Machine Learning (ICML 2012), 2012.

Richard Bellman. *Dynamic Programming*. Princeton University Press, New Jersey, 1957.

Lucian Buşoniu and Remi Munos. Optimistic planning for Markov decision processes. In *15th International Conference on Artificial Intelligence and Statistics (AI-STATS-12)*, 2012.

Herman Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *The Annals of Mathematical Statistics*, 23(4):493–507, 1952.

Thomas G Dietterich, Majid Alkaee Taleghan, and Mark Crowley. PAC optimal planning for invasive species management: improved exploration for reinforcement learning from simulator-defined MDPs. In *Association for the Advancement of Artificial Intelligence AAAI 2013 Conference (AAAI-2013)*, 2013.

Joseph M DiTomaso and Carl E Bell. *Proceedings of the Saltcedar Management Workshop*. www.invasivespeciesinfo.gov/docs/news/workshopJun96/index.html, Rancho Mirage, CA, 1996.

Eyal Even-Dar, Shie Mannor, and Yishay Mansour. PAC bounds for multi-armed bandit and Markov decision processes. In *Proceedings of the 15th Annual Conference on Computational Learning Theory*, pages 255–270, London, 2002. Springer-Verlag.

Eyal Even-Dar, Shie Mannor, and Yishay Mansour. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *Journal of Machine Learning Research*, 7:1079–1105, 2006.

Claude-Nicolas Fiechter. Efficient reinforcement learning. In *Proceedings of the Seventh Annual ACM Conference on Computational Learning Theory*, pages 88–97. ACM Press, 1994.

Claude-Nicolas Fiechter. *Design and Analysis of Efficient Reinforcement Learning Algorithms*. PhD thesis, University of Pittsburgh, Pittsburgh, PA, USA, 1997.

Irving John Good. The population frequencies of species and the estimation of population parameters. *Biometrika*, 40(3):237–264, 1953.

Rachel M. Houtman, Claire A Montgomery, Aaron R Gagnon, David E. Calkin, Thomas G. Dietterich, Sean McGregor, and Mark Crowley. Allowing a wildfire to burn: estimating the effect on future fire suppression costs. *International Journal of Wildland Fire*, 22:871–882, 2013.

Thomas Jaksch, Ronald Ortner, and Peter Auer. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11:1563–1600, 2010.

Sham M. Kakade. *On the Sample Complexity of Reinforcement Learning*. Doctoral dissertation, University College London, 2003.

Michael Kearns and Lawrence Saul. Large deviation methods for approximate probabilistic inference, with rates of convergence. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI-98)*, pages 311–319, 1998.

Michael J Kearns, Yishay Mansour, and Andrew Y Ng. A sparse sampling algorithm for near-optimal planning in large Markov decision processes. In *IJCAI*, pages 1231–1324, 1999.

Shie Mannor, O Mebel, and H Xu. Lightning does not strike twice: robust mdps with coupled uncertainty. In *Proceedings of the 29th International Conference on Machine Learning (ICML 2012)*, 2012.

David McAllester and Luis Ortiz. Concentration inequalities for the missing mass and for histogram rule error. *Journal of Machine Learning Research*, 4:895–911, 2003.

David McAllester and Robert E Schapire. On the convergence rate of Good-Turing estimators. In *Proceedings of the Thirteenth Annual Conference on Computational Learning Theory*, pages 1–6, 2000.

Rachata Muneepeerakul, Simon A Levin, Andrea Rinaldo, and Ignacio Rodriguez-Iturbe. On biodiversity in river networks: a trade-off metapopulation model and comparative analysis. *Water Resources Research*, 43(7):1–11, 2007.

Alon Orlitsky, Narayana P. Santhanam, and Junan Zhang. Always Good Turing: asymptotically optimal probability estimation. *Science*, 302(5644):427–431, 2003.

Martin Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley Series in Probability and Mathematical Statistics. Wiley, 1994.

Trey Smith and Reid Simmons. Focused real-time dynamic programming for MDPs: squeezing more out of a heuristic. In *AAAI 2006*, pages 1227–1232, 2006.

Scott M Stenquist. *Saltcedar Management and Riparian Restoration Workshop*. www.invasivespeciesinfo.gov/docs/news/workshopSep96/index.html, Las Vegas, NV, 1996.

Alexander Strehl and Michael Littman. An analysis of model-based interval estimation for Markov decision processes. *Journal of Computer and System Sciences*, 74(8):1309–1331, 2008.

Alexander L Strehl and Michael L Littman. An empirical evaluation of interval estimation for Markov decision processes. In *16th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2004)*, pages 128–135, 2004.

Umar Syed, Michael Bowling, and Robert Schapire. Apprenticeship learning using linear programming. In *International Conference on Machine Learning*, Helsinki, Finland, 2008.

Istvan Szita and Csaba Szepesvári. Model-based reinforcement learning with nearly tight exploration complexity bounds. In Johannes Fürnkranz and Thorsten Joachims, editors, *Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel*, pages 1031–1038, 2010.

Aviv Tamar, Shie Mannor, and Huan Xu. Scaling up robust MDPs using function approximation. In *ICML 2014*, volume 32, 2014.

William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3):285–294, 1933.

Gregory Valiant and Paul Valiant. Estimating the unseen: improved estimators for entropy and other properties. In *Neural Information Processing Systems 2013*, pages 1–9, 2013.

Abraham Wald. Sequential tests of statistical hypotheses. *The Annals of Mathematical Statistics*, 16(2):117–186, 1945.

Thomas J Walsh, Sergiu Goschin, and Michael L Littman. Integrating sample-based planning and model-based reinforcement learning. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*, Atlanta, GA, 2010.

Tsachy Weissman, Erik Ordentlich, Gadiel Seroussi, Sergio Verdu, and Marcelo J Weinberger. Inequalities for the L1 deviation of the empirical distribution. Technical report, HP Labs, 2003.

Journal of Machine Learning Research 16 (2015) 3905-3909 Submitted 3/14; Revised 1/15; Published 12/15

partykit: A Modular Toolkit for Recursive Partytioning in R

Torsten Hothorn TORSTEN.HOTHORN@R-PROJECT.ORG
Institut für Epidemiologie, Biostatistik und Prävention, Universität Zürich

Achim Zeileis ACHIM.ZEILEIS@R-PROJECT.ORG
Institut für Statistik, Universität Innsbruck

Editor: Cheng Soon Ong

Abstract

The R package *partykit* provides a flexible toolkit for learning, representing, summarizing, and visualizing a wide range of tree-structured regression and classification models. The functionality encompasses: (a) basic infrastructure for *representing* trees (inferred by any algorithm) so that unified `print`/`plot`/`predict` methods are available; (b) dedicated methods for trees with *constant fits* in the leaves (or terminal nodes) along with suitable coercion functions to create such trees (e.g., by *rpart*, *RWeka*, PMML); (c) a reimplementation of *conditional inference trees* (`ctree`, originally provided in the *party* package); (d) an extended reimplementation of *model-based recursive partitioning* (`mob`, also originally in *party*) along with dedicated methods for trees with parametric models in the leaves. Here, a brief overview of the package and its design is given while more detailed discussions of items (a)–(d) are available in vignettes accompanying the package.

Keywords: recursive partitioning, regression trees, classification trees, statistical learning, R

1. Overview

In the more than fifty years since Morgan and Sonquist (1963) published their seminal paper on "automatic interaction detection", a wide range of methods has been suggested that is usually termed "recursive partitioning" or "decision trees" or "tree(-structured) models" etc. The particularly influential algorithms include CART (classification and regression trees, Breiman et al., 1984), C4.5 (Quinlan, 1993), QUEST/GUIDE (Loh and Shih, 1997; Loh, 2002), and CTree (Hothorn et al., 2006) among many others (see Loh, 2014, for a recent overview). Reflecting the heterogeneity of conceptual algorithms, a wide range of computational implementations in various software systems emerged: Typically the original authors of an algorithm also provide accompanying software but many software systems, including *Weka* (Witten and Frank, 2005) or R (R Core Team, 2014), also provide collections of various types of trees. Within R the list of prominent packages includes *rpart* (Therneau and Atkinson, 1997, implementing CART), *RWeka* (Hornik et al., 2009, with interfaces to J4.8, M5', LMT from *Weka*), and *party* (Hothorn et al., 2015, implementing CTree and MOB) among many others. See the CRAN task view "Machine Learning" (Hothorn, 2014) for an overview.

All of these algorithms and software implementations have to deal with similar challenges. However, due to the fragmentation of the communities in which they are published – ranging from statistics over machine learning to various applied fields – many discussions of the algorithms do not reuse established theoretical results and terminology. Similarly, there is no common "language" for the software implementations and different solutions are provided by different packages (even within R) with relatively little reuse of code. The *partykit* aims at mitigating the latter issue by providing a common unified infrastructure for recursive partytioning in the R system for statistical computing. In particular, *partykit* provides tools for representing, printing, plotting trees and computing predictions. The design principles are:

- One 'agnostic' base class ('party') encompassing a very wide range of different tree types.
- Subclasses for important types of trees, e.g., trees with constant fits ('constparty') or with parametric models ('modelparty') in each terminal node (or leaf).
- Nodes are recursive objects, i.e., a node can contain child nodes.
- Keep the (learning) data out of the recursive node and split structure.
- Basic printing, plotting, and predicting for raw node structure.
- Customization via suitable panel or panel-generating functions.
- Coercion from existing object classes in R (rpart, J48, etc.) to the new class.
- Usage of simple/fast S3 classes and methods.

In addition to all of this generic infrastructure, two specific tree algorithms are implemented in *partykit* as well: ctree for conditional inference trees (Hothorn et al., 2006) and mob for model-based recursive partitioning (Zeileis et al., 2008).

2. Installation and Documentation

The *partykit* package is an add-on package for the R system for statistical computing. It is available from the Comprehensive R Archive Network (CRAN) at http://CRAN.R-project.org/package=partykit and can be installed from within R, e.g., using install.packages. It depends on R (at least 2.15.0) as well as the base packages *graphics*, *grid*, *stats*, and the recommended *survival*. Furthermore, various suggested packages are needed for certain special functionalities in the package. To install all of these required and suggested packages in one go, the command install.packages("partykit", dependencies = TRUE) can be used.

In addition to the stable release version on CRAN, the current development release can be installed from R-Forge (Theußl and Zeileis, 2009). In addition to source and binary packages the entire version history is available through R-Forge's *Subversion* source code management system.

Along with the package extensive documentation with examples is shipped. The manual pages provide basic technical information on all functions while much more detailed descriptions along with hands-on examples are provided in the four package vignettes. First, the vignette "partykit" introduces the basic 'party' class and associated infrastructure while three further vignettes discuss the tools built on top of it: "constparty" covers the eponymous class (as well as the simplified 'simpleparty' class) for constant-fit trees along with suitable coercion functions, and "ctree" and "mob" discuss the new ctree and mob implementations, respectively. Each of the vignettes can be viewed within R via vignette(*"name"*, package = "partykit") and the underlying source code (in R with LaTeX text) is also available in the source package.

3. User Interface

The *partykit* package provides functionality at different levels. First, there is basic infrastructure for representing, modifying, and displaying trees and recursive partitions – these tools are mostly intended for developers and described in the next section. Second, there are tools for inferring trees from data or for importing trees inferred by other software into *partykit*.

While originally an important goal for the development of *partykit* was to provide infrastructure for the authors' own tree induction algorithms CTree and MOB, the design was very careful to separate as much functionality as possible into more general classes that are useful for a far broader class of trees. In particular, to be able to print/plot/predict different trees in a unified way, there

Algorithm	Software implementation	Object class	Original reference
CART/RPart	`rpart::rpart` + `as.party`	`constparty`	Breiman et al. (1984)
C4.5/J4.8	*Weka*/`RWeka::J48` + `as.party`	`constparty`	Quinlan (1993)
QUEST	SPSS/*AnswerTree* + `pmmlTreeModel`	`simpleparty`	Loh and Shih (1997)
CTree	`ctree`	`constparty`	Hothorn et al. (2006)
MOB	`mob`, `lmtree`, `glmtree`, ...	`modelparty`	Zeileis et al. (2008)
EvTree	`evtree::evtree`	`constparty`	Grubinger et al. (2014)

Table 1: Selected implementations of tree algorithms that can be interfaced through *partykit*. The second column lists external software, R functions from other packages (with `::` syntax) and from *partykit*.

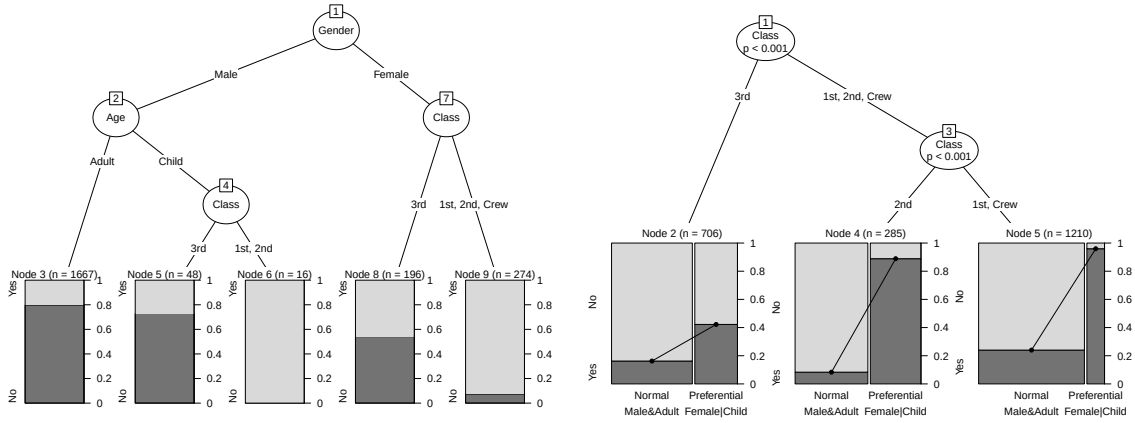


Figure 1: Tree visualizations of survival on Titanic: '`rpart`' tree converted with `as.party` and visualized by *partykit* (left); and logistic-regression-based tree fitted by `glmtree` (right).

are so-called coercion functions for transforming trees learned in other software packages (inside and outside of R) to the classes provided by *partykit*. Specifically, tree objects learned by `rpart` (Therneau and Atkinson, 1997, implementing CART, Breiman et al., 1984) and by `J48` from *RWeka* (Hornik et al., 2009, interfacing *Weka*'s J4.8 algorithm for C4.5, Quinlan, 1993) can be coerced by `as.party` to the same object class '`constparty`'. This is a general class that can in principle represent all the major classical tree types with constant fits in the terminal nodes. Also, the same class is employed for conditional inference trees (CTree) that can be learned with the `ctree` function directly within *partykit* or evolutionary trees from package *evtree* (Grubinger et al., 2014).

Not only trees learned within R can be transformed to the proposed infrastructure but also trees from other software packages. Either a dedicated interface has to be created using the building blocks described in the next section (e.g., as done for the J4.8 tree in *RWeka*) or PMML (Predictive Model Markup Language) can be used as an intermediate exchange format. This is an XML standard created by an international consortium (Data Mining Group, 2014) that includes a `<TreeModel>` tag with support for constant-fit classification and regression trees. The function `pmmlTreeModel` allows to read these files and represents them as '`simpleparty`' objects in *partykit*. The reason for not using the '`constparty`' class as above is that the PMML format only stores point predictions (e.g., a mean or proportion) rather than all observations from the learning sample. So far, the PMML interface has been tested with output from the R package *pmml* and SPSS's *AnswerTree* model. The latter includes an implementation of the QUEST algorithm (Loh and Shih, 1997).

Finally, the *partykit* function `mob` implements model-based recursive partitioning (MOB) along with "mobster" interfaces for certain models (e.g., `lmtree`, `glmtree`). These return objects of class '`modelparty`' where nodes are associated with statistical models (as opposed to simple constant fits). In principle, this may also be adapted to other model trees (such as GUIDE, LMT, or M5') but no such interface is currently available.

All of these different trees (see Table 1 for an overview) use the same infrastructure at the core but possibly with different options enabled. In all cases, the functions `print`, `plot`, and `predict` can be used to create textual and graphical displays of the tree and for computing predictions on new data, respectively. As an example for the visualizations, Figure 1 shows two different trees fitted to the well-known data on survival of passengers on the ill-fated maiden voyage of the RMS Titanic: The left panel shows a CART tree with constant fits learned by *rpart* and converted to *partykit*. The right panel shows a MOB tree learned with *partykit* with a logistic regression for treatment effects in the terminal nodes. Additionally, the are further utility functions, e.g., `nodeapply` can be employed to access further information stored in the nodes of a tree and `nodeprune` can prune selected nodes.

4. Developer Infrastructure

The unified infrastructure at the core of *partykit* is especially appealing for developers who either want to implement new tree algorithms or represent trees learned in other systems.

Here, we briefly outline the most important classes and refer to the vignettes for more details:

'`partysplit`': Split with integer ID for the splitting variable, breakpoint(s), indexes for the kids.

'`partynode`': Node specification with integer ID, a '`partysplit`', and a list of kids (if any) that are '`partynode`' objects again.

'`party`': Tree with a recursive '`partynode`' and a '`data.frame`' (optionally empty), potentially plus information about fitted values and '`terms`' allowing to preprocess new data for predictions.

All classes have an additional slot for storing arbitrary information at any level of the tree. This is exploited by '`constparty`', '`simpleparty`', and '`modelparty`' which store the observed response, point predictions, and fitted parametrics models, respectively.

5. Discussion and Outlook

Package *partykit* provides a toolkit for trees in R that gives emphasis to flexibility and extensibility. The infrastructure is easily accessible and accompanied by detailed manual pages and package vignettes. The package facilitates the implementation of new algorithms or interfacing other software by providing common building blocks for computing on trees (representation, printing, plotting, predictions, etc.). Using these building blocks developers of tree software can focus on implementing the learning algorithm (selection of variables and split points, stopping criteria, pruning, etc.). The package also provides functions for inferring trees where the computationally intensive parts are either in C (`ctree`) or employ R's fitting functions (`mob`). The simple and lean base classes that separate data and tree structure are also appealing for storing forests – a first proof-of-concept reimplementation of `cforest` is in the package with further extension planned. Users and developers that have questions or comments about the package can either contact the maintainers or use the forum on R-Forge at `https://R-Forge.R-project.org/forum/forum.php?forum_id=852`.

Acknowledgments

We are thankful to the organizers and participants of the "Workshop on Classification and Regression Trees" (March 2014), sponsored by the Institute for Mathematical Sciences of the National University of Singapore, for helpful feedback and stimulating discussions.

References

Leo Breiman, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone. *Classification and Regression Trees*. Wadsworth, California, 1984.

Data Mining Group. Predictive model markup language, 2014. URL `http://www.dmg.org/`. Version 4.2.

Thomas Grubinger, Achim Zeileis, and Karl-Peter Pfeiffer. *evtree*: Evolutionary learning of globally optimal classification and regression trees in R. *Journal of Statistical Software*, 61(1), 1–29 2014.

Kurt Hornik, Christian Buchta, and Achim Zeileis. Open-source machine learning: R meets *Weka*. *Computational Statistics*, 24(2):225–232, 2009.

Torsten Hothorn. CRAN task view: Machine learning & statistical learning, 2014. URL `http://CRAN.R-project.org/view=MachineLearning`. Version 2014-12-18.

Torsten Hothorn, Kurt Hornik, and Achim Zeileis. Unbiased recursive partitioning: A conditional inference framework. *Journal of Computational and Graphical Statistics*, 15(3):651–674, 2006.

Torsten Hothorn, Kurt Hornik, Carolin Strobl, and Achim Zeileis. *party: A Laboratory for Recursive Partytioning*, 2015. URL `http://CRAN.R-project.org/package=party`. R package version 1.0-20.

Wei-Yin Loh. Regression trees with unbiased variable selection and interaction detection. *Statistica Sinica*, 12(2):361–386, 2002.

Wei-Yin Loh. Fifty years of classification and regression trees. *International Statistical Review*, 82 (3):329–348, 2014.

Wei-Yin Loh and Yu-Shan Shih. Split selection methods for classification trees. *Statistica Sinica*, 7 (4):815–840, 1997.

James N. Morgan and John A. Sonquist. Problems in the analysis of survey data, and a proposal. *Journal of the American Statistical Association*, 58(302):415–434, 1963.

John R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, 1993.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2014. URL `http://www.R-project.org/`.

Terry M. Therneau and Elizabeth J. Atkinson. An introduction to recursive partitioning using the *rpart* routine. Technical Report 61, Section of Biostatistics, Mayo Clinic, Rochester, 1997. URL `http://www.mayo.edu/hsr/techrpt/61.pdf`.

Stefan Theußl and Achim Zeileis. Collaborative software development using R-Forge. *The R Journal*, 1(1):9–14, May 2009.

Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, San Francisco, 2nd edition, 2005.

Achim Zeileis, Torsten Hothorn, and Kurt Hornik. Model-based recursive partitioning. *Journal of Computational and Graphical Statistics*, 17(2):492–514, 2008.